WP3 – Adapt-&-Play Seamless Integration of Legacy Equipment and Building Systems

Document Version: **1.0**

D3.2 Technical upgrades and integration mechanism for legacy equipment – Intermediate version

Project Number:	Project Acronym:	Project Title:
893079	PHOENIX	Adapt-&-Play Holistic cOst-Effective and user-frieNdly Innovations with high replicability to upgrade smartness of eXisting buildings with legacy equipment

Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type* - Security**:
31/12/2021	31/12/2021	R – PU

* Type: P – Prototype, R – Report, D – Demonstrator, O – Other

** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	**Organization:**	**Contributing WP:**
Fergal Purcell	ARDEN	WP3

Authors (organizations):

ARDEN, UMU, MIWENERGIA, KaMa, OdinS, UBITECH, SAGOE and LTU

Abstract:

This document describes the mechanism for integration of legacy equipment performed within the four pilots: Greece, Spain, Sweden and Ireland. The strategies for increasing connectivity include the installation of new hardware devices as well as middleware, the replacement of systems and the software connection of existing devices to the general platform. This document is a second version of a catalogue of integration mechanisms that will form an output from work package 3.

Keywords:

Smartness; Integration; Legacy equipment; Communication; Middleware.

Revision History

The following table describes the main changes in the document since created.

Revision	Date	Description	Author (Organization)
0.1	26/11/2021	Preparation of the skeleton of the document	ARDEN
0.2	10/12/2021	Chapter 6: Input for the Greek pilot	Eirini Christoulaki (KAMA)
0.3	14/12/2021	Description of Irish Pilot and BMS communications	ARDEN
0.4	16/12/2021	Data models and SRI	UMU
0.5	17/12/2021	Integration of legacy devices	ODINS
0.6	18/12/2021	Review and editing	Arden
0.7	22/12/2021	Revised document	Eirini Christoulaki (KAMA)
0.8	23/12/2021	Description of the Swedish Pilot	LTU
0.9	23/12/2021	Document review	VERD
1.0	28/12/2021	Final Document	ARDEN

Executive Summary

This document is the second part of a set of deliverables that describes the mechanisms of integration of legacy equipment within the PHOENIX platform for the duration of the project. Building on the first, D3.1, the document describes the upgrades and integration of legacy equipment being carried out as the PHOENIX project evolves. There is a description of the integration methods utilised to connect newly and previously installed devices on site to connect to the PHOENIX platform at the pilot sites. Any upgrades are described as well as any difficulties encountered during integration.

Also included in the document is a description of the external data sources connected to the PHOENIX platform. These include the electricity market data and weather data of pilot countries.

The use cases for each pilot are described as well as how they have so far been applied and how the integration of the legacy devices at each site relates to these use cases. All systems at each pilot site are detailed and their connectivity is explained.

The methods for which the devices are integrated to the PHOENIX platform are described in detail in this document. Each abstraction level for different types of data collection and actuation are conveyed for Z-Wave devices and Modbus. The relation between device integration and the SRI impact at each pilot site is described.

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 893079, but this document only reflects the consortium's view. The European Commission is not responsible for any use that may be made of the information it contains.

Table of Contents

1. Introduction

1.1. Scope of the Document

This document is the Intermediate version of a set of three deliverables that describe the upgrades and integration of legacy equipment implemented as the project PHOENIX evolves. This intermediate version includes updates on the integration at pilot sites thus far. This relates to the installation and integration of legacy equipment via the creation of novel gateway-solutions and the connection to quasi-smart devices thanks to middleware that connects to different communication mechanisms. In addition, this deliverable includes the services planned for each pilot site following full integration of equipment as well as the SRI score achieved after each successful integration.

1.2. Relevance to other deliverables

This deliverable is related to the rest of the set namely Deliverable 3.1 and 3.3. The integration methods adhered to this deliverable were first developed in the POC in Deliverable 3.2 and will be fully developed in Deliverable 3.3. This deliverable is also relevant to D7.2 “Initial Pilots Deployment, Operation and Validation”. In this document, how each pilot leader deployed their legacy equipment, enhanced operation and socioeconomic impacts is described.

1.3. Structure of the Document

This document includes twelve sections including this introduction. Section 2 describes the BMS integrations conducted at pilot locations in Ireland and Spain. Section 3 describes the integration of external data sources from electricity markets and also metering systems. Sections 4 to 8 provide a description of the pilot sites; legacy devices are described in detail and installed or enhanced devices are listed along with their SRI impact. Furthermore, the systems available at each pilot site were described along with details about how they were integrated to the PHOENX platform. Section 9 contains detailed descriptions of how the legacy devices were integrated with the PHOENIX platform via Modbus and Z-Wave. Following on, Section 10 details the PHOENIX platform components and data modelling. Section 11 conveys the SRI framework within the PHOENIX project. Section 12 contains all references used in this document.

2. BMS Integration

2.1. Spanish Pilot

Two major changes have been made in the integration of the Pleiades building BMS of the UMU Pilot:

- Actuation support.
- Improvements to reduce the lag detected in certain situations.

The actuation support has been implemented using the Context Provider concept offered by the Context Broker. This is described in detail later in this document, in the section related to the Actuation agent for Z-Wave and Intesis WMP devices.

Once the actuation support was added, there are certain situations in which the delay between sending an actuation and getting the status of the devices updated was not acceptable. The most obvious case is the air conditioning system, where long delays might interfere the normal operation of certain services related to air quality or comfort, for example. These delays can be even bigger as the 5 split system air conditioning units BMS, and the access to all the splits of the building is cyclic, so reading or writing values to one of them requires the control process of the gateway to reach this specific one, which might take some time.

Given how the BMS works with this type of devices, the connector has been updated to work with the splits asynchronously in order to send the read/write requests faster, but only for these connectors as it's still working synchronously with all the other devices.

2.2. Irish Pilot

The Arden commercial pilot is the National Centre for the Circular Economy, a state of the art facility with a focus on sustainability and reuse [5]. The facility has a range of energy appliances and technologies which are controlled by an existing BMS system. As part of the Phoenix project the BMS was upgraded with Enteliweb software to facilitate API connectivity with the BMS. The Enteliweb software allows read and write access via API queries and messages in XML or json format. The general format of XML queries is

http://<server IP address>/enteliweb/api/.bacnet/<site name>.

Middleware was developed to periodically query the Enteliweb BMS API and to reformat and send as MQTT json files to the Phoenix context broker. The process required for setting data is slightly more complex, firstly the data type of the object you are setting must be determined. This can be found by sending a get request for the property of the device to which you are looking to set a value.

Format: *GET http://<server IP address>/enteliweb/api/.bacnet/<site name>/<device number>/< object type>,<instance>/<property name>/<sub-property path>*

The format and example request and response of a put request of this type is outlined below.

Format : *PUT http://<server IP address>/enteliweb/api/.bacnet/<site name>/<device number>/< object type>,<instance>/<property name>/<sub-property path>* *Body : <<datatype> xmlns="http://bacnet.org/csml/1.2" value="<value>"/>* *Response :* *<Enumerated error="-1" errorText="OK" xmlns="http://bacnet.org/csml/1.2"/>*

The middleware developed by Arden listens for messages from the Phoenix context broker for actuation and then mediates between the Enteliweb API and the Phoenix context broker in transmitting data, actuation signals and responses.

3. External data sources Integration

3.1. Entsoe

Extracted verbatim from its website, *ENTSO-E is the European association for the cooperation of transmission system operators (TSOs) for electricity*, and its mission is to *ensure the security of the interconnected power system in all time frames at pan-European level and the optimal functioning and development of the European interconnected electricity markets, while enabling the integration of electricity generated from renewable energy sources and of emerging technologies*.

In the context of the PHOENIX project, Entsoe is an external data source that provides information of the grid at a country level in Europe and therefore offers various resources to grid-oriented services in WP6.

Among the information provided by Entsoe, five sets of values have been selected as the most relevant and have been integrated in the platform (sets of readings per country, not per pilot):

- Total load. Hourly readings of the total load during the previous day.
- Total load day-ahead. Hourly readings of the forecasted total load for the next day.
- Generation per type. Hourly readings of the generation by source type during the previous day (nuclear, solar, etc.).
- Wind and solar generation day-ahead. Hourly readings of the forecasted wind and solar generations for the next day.
- Day-ahead prices. Hourly (well-known and official) prices for the next day.

The way to access to all this information is through a REST API available at the Entsoe Transparency Portal [2]. Once an authorization key has been obtained (free registration), the data can be downloaded with certain restrictions in terms of the number of requests that can be sent each day.

The integration is done using an instance of Node-Red that’s being executed in the PHOENIX platform itself. It includes one main flow that’s in charge of sending the requests and a subflow that manages each individual request.

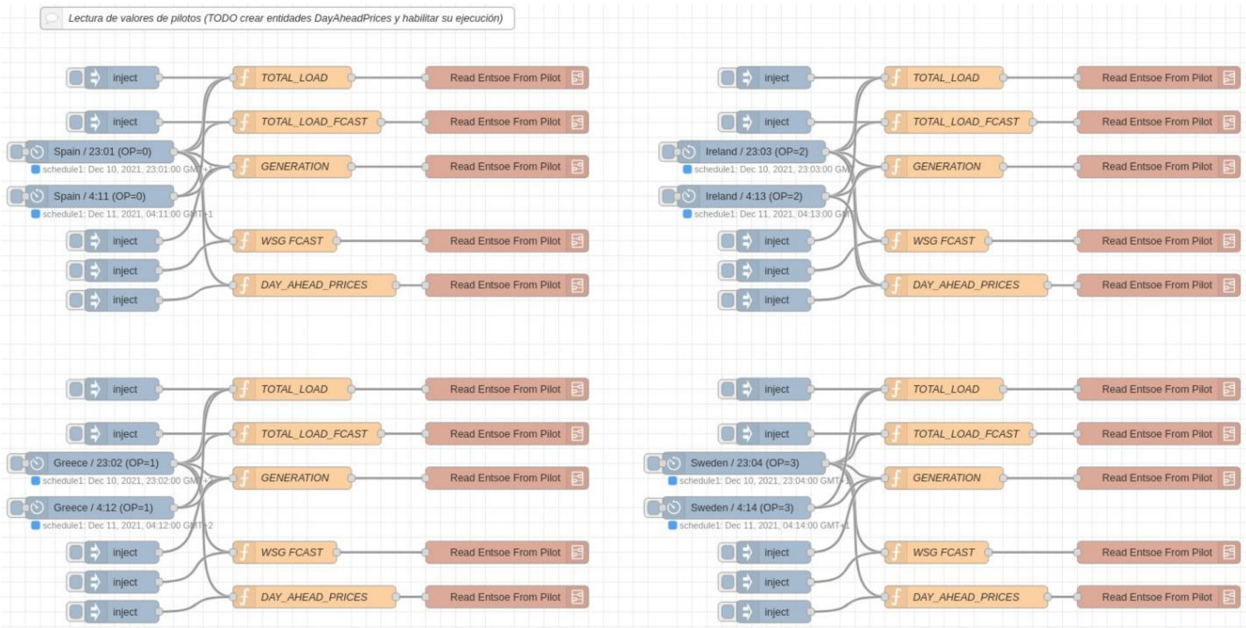


Figure 1. Entsoe / Main flow

Some of the values are only obtained in a night reading round (close to 23:00 local time for each country) while there's an extra reading round in the morning (close to 4:10 also local time for each country) added in an attempt to get certain values that are not available in the night reading round in some cases.

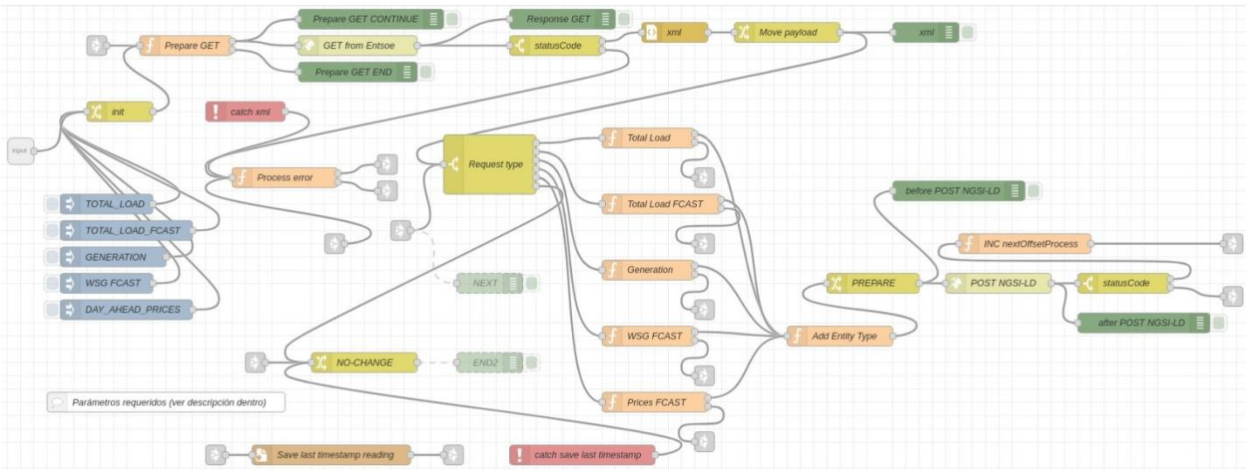


Figure 2. Entsoe / Individual reading subflow

The subflow is designed to send a reading request based on the parameters received from the main flow, which indicate whether to read values from yesterday, today or tomorrow (previous readings, day-ahead readings, etc.), type of reading, etc., process the response and send the received values to the Context Broker. The timestamp of the last reading is saved in disk once it has been sent to the Context Broker in order to know where to start next time in case the whole agent must be restarted for some reason. A sample request to get day-ahead prices from Spain in a specific time range is this:

```
curl -X GET https://transparency.entsoe.eu/api?securityToken=token-received&documentType=A44&in_Domain=10YES-REE------0&out_Domain=10YES-REE------0&timeInterval=2021-10-10T23:00Z/2021-10-11T23:00Z
```

And the most relevant part of the response to the previous request (xml format) is this:


```
<Period>
 <timeInterval>
  <start>2021-10-11T22:00Z</start>
  <end>2021-10-12T22:00Z</end>
 </timeInterval>
 <resolution>PT60M</resolution>
 <Point>
  <position>1</position>
  <price.amount>176.13</price.amount>
 </Point>
 <Point>
  <position>2</position>
  <price.amount>175.53</price.amount>
 </Point>
...
 <Point>
  <position>24</position>
  <price.amount>176.13</price.amount>
 </Point>
</Period>
```


3.2. REE

Red Eléctrica Española (REE) is the sole transmission system operator (TSO) of the national electricity system in Spain. REE establishes the forecasts of the demand for electricity and operates the electricity generation and transmission facilities in real time.

Therefore, REE gathers a large amount of information about Spanish electrical system status and, according to current legislation, REE is under the obligation to publish details of electrical market results or system operating processes. Regarding REE's role within PHEONIX, REE is an external data source that provides information about electricity tariffs for each hour of the day. Data about electrical market prices are especially valuable to grid-oriented services from WP6.

Everyone can access to all this information through a REST API provided by REE [3]. To perform the integration of REE external data source, a Node-Red flow periodically requests data to REE's

REST API. At the beginning of each hour, it makes a request to the API and retrieves the electricity tariff for that specific hour. This information is mapped into a dedicated Context Broker entity and it is sent to the Context Broker. If the dedicated entity already exists in the Context Broker, it is updated with the new electricity price values. Otherwise, a new entity is created with the obtained values. The next figure shows Node-Red flow.

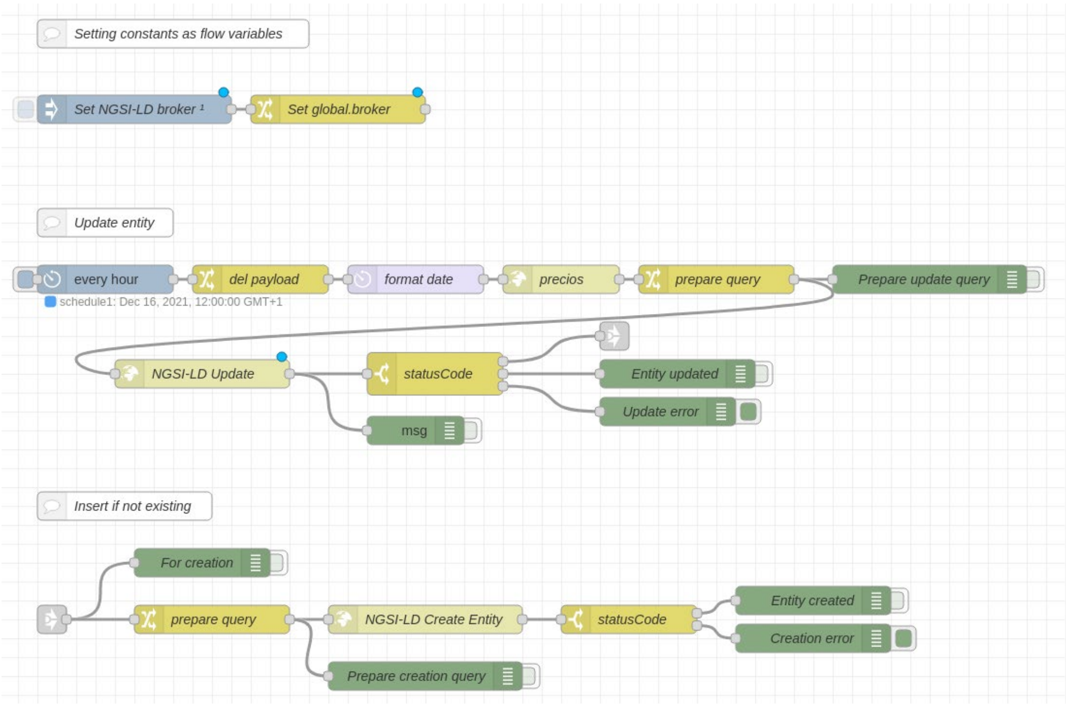


Figure 3. REE electrical tariff prices flow

This is a sample request to get electric market prices in a specific time range (one hour specifically):

```
curl -X GET
https://apidatos.ree.es/es/datos/mercados/precios-mercados-tiempo-real?start_date=2021-10-15T12:00&end_date=2021-10-15T12:59&time_trunc=hour&cached=true
```

And this is the most useful part of the response to the previous request:


```
{
  "type": "PVPC (€/MWh)",
  "id": "1001",
  "groupId": null,
  "attributes": {
    "title": "PVPC (€/MWh)",
    "description": null,
    "color": "#ffcf09",
    "type": null,
    "magnitude": "price",
    "composite": false,
    "last-update": "2021-10-14T20:17:02.000+02:00",
    "values": [
      {
        "value": 288.94,
```

```
        "percentage": 0.57140034,
        "datetime": "2021-10-15T12:00:00.000+02:00"
      }
    ]
  }
}
```

3.3. UMU Electricity metering at building level

UMU has a SCADA system where the consumption of several buildings is being stored. This SCADA is directly connected to different power meters from most of the buildings and reads them periodically. Analysis of this data allows improvement of some processes such as switching HVAC systems on in a smart way to reduce consumption peaks having both a local (building) and a global (campus) view of the consumption, for example. UMU provides access to this information through a REST API.

The integration also uses the instance of Node-Red that's being executed in the PHOENIX platform itself. It includes one main flow that's in charge of sending the requests and a subflow that handles each individual request.

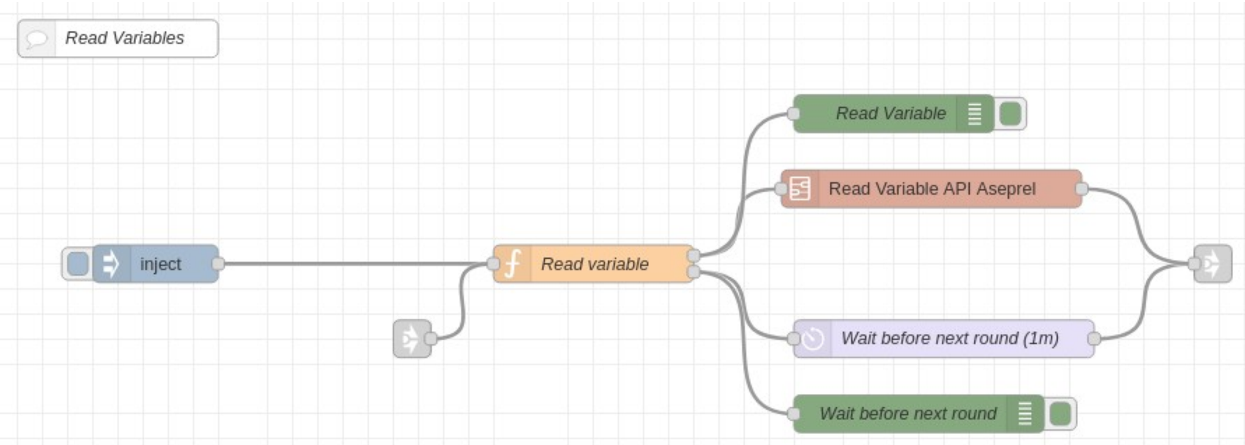


Figure 4. UMU Electricity metering at building level / Main flow

The reading of variables is scheduled so that a minimum of a one-hour delay is maintained between the moment the request is sent and the timestamp of the expected value. It must be done this way because the SCADA is updated every hour and it will take time before the readings for the current time are available.

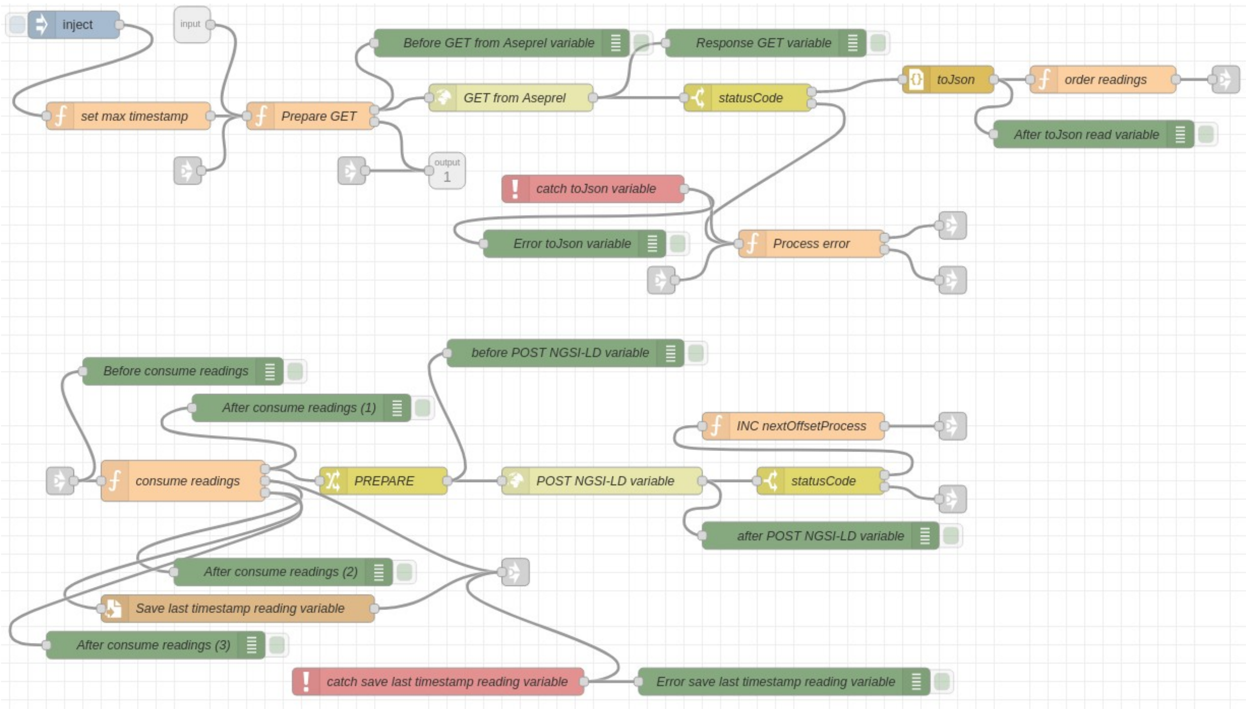


Figure 5. UMU Electricity metering at building level / Subflow

Again, the timestamp of the last reading is saved in disk once it is sent to the Context Broker to determine where to start next time in case the whole agent must be restarted for some reason.

3.4. UMU EV chargers

The UMU SCADA also provides access to real time and historical data of existing EV chargers in a similar way but through a different endpoint.

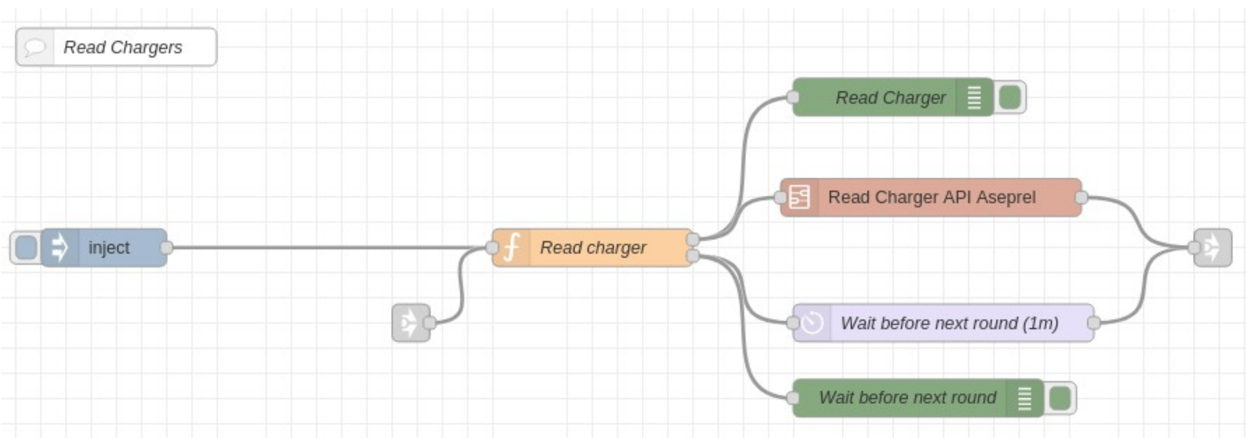


Figure 6. UMU EV chargers / Main flow

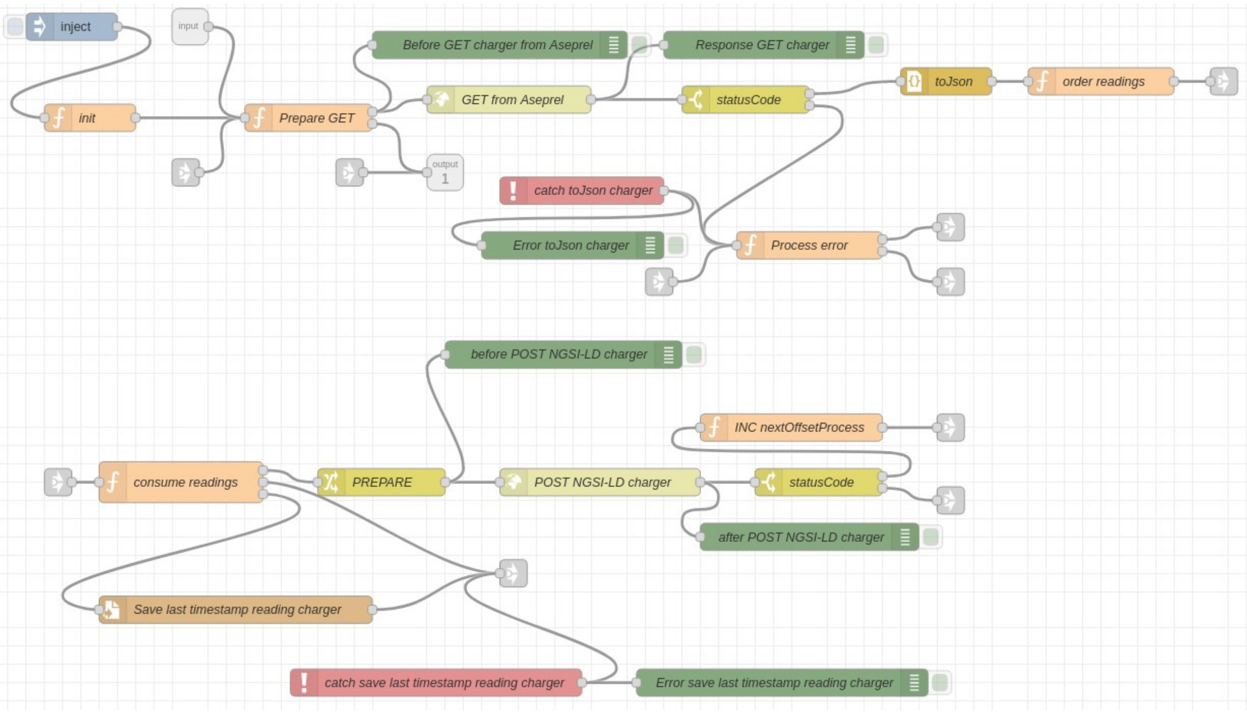


Figure 7. UMU EV chargers / Subflow

No information of the API format is provided as it belongs to UMU.

4. UMU Pilot

4.1. Description

An important part of the UMU pilot was already included in D3.1 as this is the pilot used for the PoC of the project. However there are other elements that have been either installed or integrated after the PoC was finished.

4.2. Use Cases

UC01: Adapt & Play integration of domestic appliances, legacy equipment and building systems

The UMU pilot has served to plan the integration of a variety of devices that need to be adapted to be connectable and controllable via the PHOENIX platform. Standard protocols such as Modbus have been identified as common place to make the legacy devices able to connect to the PHOENIX platform.

UC04: Provision of Comfort, Convenience and Wellbeing services to building occupants

Within the devices that will need to be integrated on the PHOENIX pilot we find a series of sensors that are oriented to comfort, convenience and wellbeing. Examples of these are temperature sensors, humidity sensors and luminosity sensors that are now in the form of legacy equipment connected to an existing BMS. This is commonplace on existing buildings. The use case for UMU on this regard will be to provide sufficient connectivity to this BMS to make it part of PHOENIX.

UC06: Flexible billing services and smart contracts for the retailer customers

As mentioned before, the general UMU campus offers the opportunity of testing the consumers within it as an energy community. For this, the use case on this regard for the UMU pilot will be to create a digital twin of the movement of energy, and once calibrated and validated, the testing of the billing mechanisms and the smart contracts will be evaluated. Conversations with the energy utility will be established to transfer to them the lessons learnt and to identify ways of making more advanced billing mechanisms or price schemas. This plan goes in accordance with the project as a whole, as UMU is the leader of WP6 what covers grid flexibility.

4.3. Legacy equipment and systems

The list of additional devices and systems of the pilot is detailed in Table 1.

Table 1 – Legacy Devices at the UMU Pilot Site

Details of equipment (e.g. capacity, model, power, etc.)	Use Cases
Monitoring devices (electrical consumption): Multiple power meters in different buildings of Campus de Espinardo	UC1, UC6
Solar hot water (electrical consumption): CVM-Mini power meters for additional resistors in both SOLAR and DHW tanks (extra elements for this system)	UC1, UC6
Monitoring devices (generation): Gateway + Communication interface + SW integration to monitor the backup generator of the Pleiades building	UC1, UC6
EV charging point: Integration of EV charging point at the Computer Science Faculty's parking	UC1, UC6

4.4. Systems

4.4.1. Backup generator of Pleiades

In one of the sheds adjacent to the Pleiades building, next to the one where the Solar DHW system is installed, there's a backup generator. This diesel generator is used for two purposes:

- Provide energy in case of power cut.
- Reduce the peaks of energy consumption twice a year when the global climate systems are switched from winter to summer mode and vice-versa.

Figure 8. Pleiades backup generator

Hardware and software connectivity

- The generator now has Modbus-RTU support through a communication card.
- An additional Modbus/TCP gateway opens the access to the generator to a TCP/IP network.
- The controller responsible of the Solar DHW system integration is also monitoring the generator through this gateway.
- EV charger of Computer Science Faculty's parking. In the parking of the Computer Science Faculty there's an EV charger.

Figure 9. EV charger

As described in the section related to External data sources Integration, the information of the charger is retrieved from the UMU Scada.

5. ARDEN Pilots

5.1. Description

The ARDEN pilot is formed by commercial and residential buildings, having a large wealth of data. The commercial pilot building is the National Centre for the Circular Economy in Ireland, the Rediscovery Centre. The building is a repurposed boiler house and includes solar PV, combined heat and power (CHP), heat pump and solar thermal. With the successful installation of middleware, data from the Building Management System is being sent to the PHOENIX platform. This data relays energy consumption from the various generators and devices on site as well as air quality data.

In relation to the domestic sites, all hardware has been installed. ARDEN has successfully obtained an API for both home energy systems and middleware has been developed and data will soon be routinely sent to the PHOENIX platform.

5.2. Use Cases

The use cases that describe the Irish pilots' functionalities supported by the PHOENIX framework are the following:

<u>UC01: Adapt & Play integration of domestic appliances, legacy equipment and building systems</u>

This use case will be implemented in all pilot sites. An existing BMS system has been integrated with Phoenix at the commercial pilots and various legacy devices and systems will be integrated at the domestic sites.

<u>UC02: Building knowledge enhancement to upgrade the smartness of buildings</u>

At the commercial pilot, the fully integrated BMS provides historical data and enables load switching. At the domestic pilot sites residents will be provided with information on consumption and energy performance as well as enhanced controls.

<u>UC03: Services for building occupants to maximize their energy efficiency and increase overall building performance</u>

Similarly to UC02, upgrades and integration with Phoenix will allow monitoring buildings' energy performance and promotion of energy savings as well as providing controls to optimise efficiency.

<u>UC04: Provision of Comfort, Convenience and Wellbeing services to building occupants</u>

The enhanced smart controls and information provided to building managers and occupants at the Irish pilots will improve comfort and convenience. Room thermostats and smart controls will ensure that rooms are heated to the desired temperatures at the correct times. Hot water controls will make hot water available when needed at the lowest possible cost.

UC05: Portfolio flexibility analysis and configuration to optimize grid operation

Demand response and flexibility for grid optimisation will be enabled through predictive control response of significant electrical loads facilitating load shifting according to time of use pricing. In the commercial pilot operation of the heat pump and CHP unit will be optimised according to ½ hourly electricity prices.

UC06: Flexible billing services and smart contracts for the retailer customers

With the utilisation of predictive control response and the implementation of time of use charges in Ireland, the pilots will be provided with a lot more flexibility in controlling their consumption and costs. Arden Energy will issue bills based in hourly time of use pricing incorporating a comparison to standard bills and savings to the site achieved through load shifting and grid flexibility.

UC07: Advanced energy services to promote self-consumption optimization

The combination of user information, enhanced control and information on electricity pricing signals will promote the optimisation of self-consumption of on-site generation and minimisation of grid imported electricity.

5.3. Legacy devices and systems

The legacy devices and systems at the Irish commercial pilots are detailed in Table 2 following. The site has a range of heat producing appliances including a gas fired boiler, a micro combined heat and power (CHP) unit, solar thermal, an air source heat pump and a wood stove.

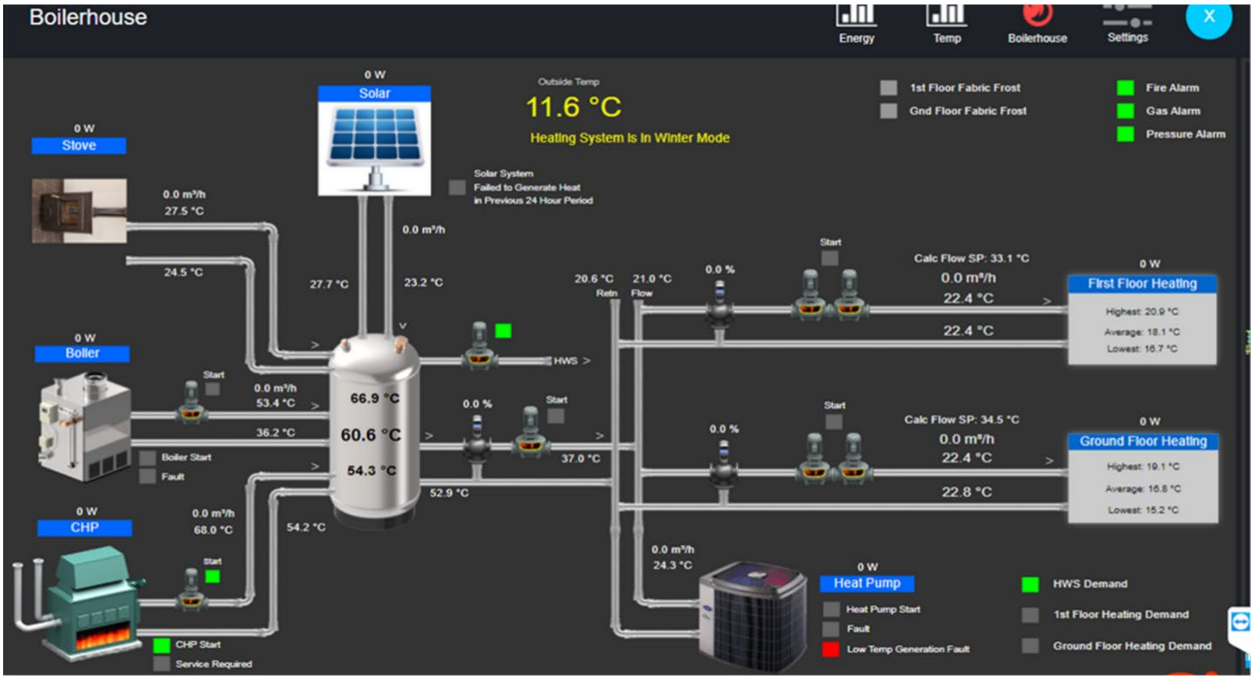


Figure 10. – Legacy equipment at the Rediscovery Centre Irish Pilot

The equipment is connected and controlled via an existing Building Management System (BMS) on site. The BMS was upgraded with Enteliweb software an API module to allow for connectivity with the Phoenix platform, as described in Section 2.2. The API allows for both read and write access.

Table 2. Legacy Devices at the Irish Commercial Pilot Site

Details of equipment (e.g. capacity, model, power etc)	Use Cases
Heating system: CHP - Dachs/Senertech G5.5 Generation 1.1 5.5kWe 14.7kWth Condensing 30kW - Manufacturer: Remeha Model: Qunita Pro 30 Heat pump: Neura 20kW Model: L20EuC	UC01, UC04, UC05
LED- Lighting: Various types of led lamps and bulbs	UC04
DHW: As for heating plus solar thermal boilers with differential thermostat	UC01
Renewable energy: Solar PV panels and inverters: GROWatt 2500 (2.5 kW) and Solar Edge SE5K (5 kW)	UC01, UC05, UC06
Monitoring and Control: Delta control BMS with EnteliWeb - Energy Management Software	All
Monitoring devices (electrical consumption/ generation): Three phase smart meter on main incomer and 18 sub metered loads	UC01, UC02, UC03, UC05, UC06
Monitoring devices (temperature): 16 x 10K3A1 -40-150 °C AIC connected to BMS	UC01, UC03, UC04, UC05, UC06, UC07
Monitoring and control Building Management System. Delta Controls/Enteliweb	UC01, UC03, UC04, UC05, UC06, UC07

Table 3 following details the legacy equipment and systems at the domestic pilot sites. Both sites were provided with a MyEnergi home gateway for connectivity as part of the Phoenix project. The MyEnergi gateway connects outputs from electricity meters, EV chargers, hot water boilers and solar PV via and API and allows for read and write aces and for optimisation of usage patterns.

Table 3. Legacy Devices at the Irish Commercial Pilot Site

Domain	Legacy Equipment and Systems
Heating system	Gas fired boiler
Domestic hot water	Electric and gas fired
Electricity renewables and storage	Solar PV
EVs equipment	EV charging

5.4. Interventions and Requirements

The interventions planned in the Irish pilot include energy savings, grid flexibility and comfort and health. In order to deliver these interventions and services there are specific requirements in terms of integration of legacy devices with the Phoenix platform for remote monitoring and control. In general connectivity between sensors, meters and energy using and generating appliances is needed to deliver the Phoenix services and write access is needed in some cases, e.g. for EV charging, heat pump operation and electric hot water boiler operation.

Table 4. Services planned in the Irish Pilot

Field	Service	Requirement
Energy savings	Improved control	Integration of controls on heating and hot water for write controls from Phoenix.
Grid flexibility	Load shifting	Integration of controls on heating and hot water for write controls from Phoenix. Integration of electricity market data.
Comfort and Health	Ambient temperatures and CO_2	Integration of ambient sensors with Phoenix platform for visualisation and alerts.
Information to occupants	Monitoring and Control	Integration of sensors and plant status with Phoenix platform for visualisation.
Grid flexibility	Optimisation of EV charging	Integration of controls on EV charging for write controls from Phoenix. Integration of electricity market data.
Self-generation	Solar PV self-generation	Integration of solar PV generation data in Phoenix with forecasting and load shifting to improve self-consumption.

These services will lead to improved smartness readiness and a quantified improvement in the SRI in the pilot sites under various domain smart readiness indicator categories. There are specific hardware and connectivity requirements to enable the delivery of these smart services and the improvement in smart readiness under the specific domain. Table 5 lists the specific SRI improvements achieved in the Irish pilots together with the pilot location and SRI point gains through the specific connectivity measure. In most cases the costs of connectivity in the Irish pilots are aggregated and combined in the cost of BMS upgrades, gateway installation and middleware development so cannot always be broken out by specific connectivity improvement.

Table 5. Interventions planned in the Irish Pilots

Domain SRI code	Hardware and connectivity	Pilot Location	SRI points gained
Heating	Individual room control with communication between controllers and to BACS	Commercial Pilot	2
Heating	Variable temperature control depending on outdoor temperature	Commercial Pilot	2
Heating	HW storage vessels controlled based on external signals (from BACS or grid)	Commercial Pilot	1
Heating	Central or remote reporting of performance evaluation including forecasting and/or benchmarking	Commercial Pilot	2
DHW	Automatic on/off control, scheduled charging enable and demand-based supply temperature control or multi-sensor storage management	Commercial Pilot	2
DHW	Automatic charging control based on local availability of renewables or information from electricity grid (DR, DSM)	Commercial Pilot	3
DHW	Performance evaluation including forecasting and/or benchmarking	Commercial Pilot	2
Ventilation	Central Demand Control based on air quality sensors (CO_2, VOC...)	Commercial Pilot	7
Ventilation	Real time monitoring & historical information of IAQ available to occupants	Commercial Pilot	2
Lighting	Automatic detection (manual on / dimmed or auto off)	Commercial Pilot	1
Electricity	Real-time feedback or benchmarking on appliance level	Commercial Pilot	5
Electricity	Performance evaluation including forecasting and/or benchmarking	Commercial Pilot	2
Monitoring and Control	Single platform that allows automated control & coordination between TBS + optimization of energy flow based on occupancy, weather and grid signals	Commercial Pilot	2
Heating	Variable temperature control depending on outdoor temperature	Domestic #1	2
Heating	HW storage vessels controlled based on external signals (from BACS or grid)	Domestic #1	1
Heating	Central or remote reporting of performance evaluation including forecasting and/or benchmarking	Domestic #1	5
DHW	Automatic charging control based on local availability of renewables or information from electricity grid (DR, DSM)	Domestic #1	3
DHW	Performance evaluation including forecasting and/or benchmarking	Domestic #1	5
Lighting	Manual on/off switch + additional sweeping extinction signal	Domestic #1	3
Electricity	On site storage of energy (thermal storage) with controller optimising the use of locally	Domestic #1	4

	generated electricity		
Electricity	Real-time feedback or benchmarking on appliance level	Domestic #1	6
Electricity	Performance evaluation including forecasting and/or benchmarking	Domestic #1	5
EV Charging	Reporting information on EV charging status to occupant	Domestic #1 & #2	3
Monitoring and Control	Single platform that allows automated control & coordination between TBS	Domestic #1 & #2	4
Monitoring and Control	Coordinated demand side management of multiple TBS	Domestic #1 & #2	5
Monitoring and Control	Central or remote reporting of real time energy use per energy carrier, combining TBS of at least 2 domains in one interface	Domestic #1 & #2	7
DHW	Automatic control on / off and scheduled charging enable	Domestic #1 & #2	3
DHW	Actual values and historical data	Domestic #1 & #2	1
Electricity	Real-time feedback or benchmarking on building level	Domestic #1 & #2	3
EV Charging	1-way controlled charging (e.g. including desired departure time and grid signals for optimization)	Domestic #2	5
Monitoring and Control	Demand side management possible for individual TBS, but not coordinated over various domains	Domestic #2	2
Monitoring and Control	Central remote reporting of real time energy use per energy carrier	Domestic #2	4

5.5. Systems

5.5.1. Rediscovery Centre

In 2018, the Rediscovery Centre had a Building Management System installed by ControlTech. The BMS allows control and monitoring of the building's mechanical and electrical equipment such as the Solar PV, ventilation and lighting. It records five-minute data points for each device. Currently the BMS is not being utilised to its full capabilities as the computer which controls it is not in the relative vicinity to where employees work. In order to enable connection to the PHOENIX platform, and thus easy access for employees to the BMS, middleware had to first be installed. Data is now being sent to the PHOENIX platform. This software (Building System Adapter) is running in the cloud with required connectivity and capacity and is sending this information to the PHOENIX hub using a MQTT interface. Once fully operational, this will allow occupants control over the various devices on site through their phone or laptop.

Measuring and actuation capacities

The Rediscovery Centre has a Delta Controls BMS system which allows for monitoring of ambient conditions and scheduling and control of plant. There are 10 indoor temperature sensors, 5 CO_2 and VOC sensors and a range of heat and electricity meters. All the plant is controlled via the BMS. Table 6 lists the measuring devices available, all of which have been integrated with the Phoenix platform.

Figure 11. BMS System

Hardware and Software connectivity

The Enteliweb software provides a wide array of API functions, the comprehensive documentation of these function is listed in the resources section of this folder. Annex 1 provides a detailed description of how to extract data from sensors set up using the Enteliweb Building Management Software (BMS).

5.5.2. Domestic Pilots

With respect to the two domestic pilots, ARDEN has installed new hardware that was selected with the purpose of easing the connectivity to the PHOENIX platform. The MyEnergi home portal solution will be used at both sites (one site already has a MyEnergi hub and one will be installed at the other site), work has already begun on connecting the site with the MyEnergi portal already active. The MyEnergi API is being utilised to connect the hubs with the PHOENIX platform via middleware which is being developed to run on the cloud to send data to the PHOENIX platform using a MQTT interface.

Figure 12. Domestic MyEnergi gateway installed

<u>Measuring and actuation capacities</u>

- Main incoming electricity supply
- Water heater consumption
- EV consumption
- Solar power generation
- EV charging actuation
- Water heater actuation

Figure 13. Solar inverter and electricity meter

Hardware and Software connectivity

- Wired connection to gateway from meters and actuators.
- Wi-Fi transmission from gateway to third party platform and API
- Middleware and integration with PHOENIX MQTT IoT-Agent)

5.6. Integration and Data Validation

Table 6 ARDEN devices integrated in commercial pilot

Device Type	**Device ID**	**Connectivity**
Temperature Sensor	Outside Temp	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 1	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 2	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 3	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 4	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 5	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 6	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 7	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 8	BMS & Middleware
Temperature Sensor	Garden_Temp_Gnd 9	BMS & Middleware
Temperature Sensor	Room_Temp_Gnd 10	BMS & Middleware
Air Quality - CO2	Room_Co2 1	BMS & Middleware
Air Quality - CO2	Room_Co2 2	BMS & Middleware
Air Quality - CO2	Room_Co2 3	BMS & Middleware
Air Quality - CO2	Room_Co2 4	BMS & Middleware
Air Quality - CO2	Room_Co2 5	BMS & Middleware
Air Quality – VOC	Room_TVOC 1	BMS & Middleware
Air Quality – VOC	Room_TVOC 2	BMS & Middleware
Air Quality – VOC	Room_TVOC 3	BMS & Middleware
Air Quality – VOC	Room_TVOC 4	BMS & Middleware
Air Quality – VOC	Room_TVOC 5	BMS & Middleware
Temperature Sensor	Garden_Temp_Gnd 11	BMS & Middleware
Heat Meter	HeatPump HM_ENERGY WATT-HR	BMS & Middleware
Heat Meter	CHP HM_ENERGY WATT-HR	BMS & Middleware
Heat Meter	Boiler HM_ENERGY WATT-HR	BMS & Middleware
Heat Meter	Solar HM_ENERGY WATT-HR	BMS & Middleware
Heat Meter	Ground HM_ENERGY WATT-HR	BMS & Middleware
Heat Meter	First HM_ENERGY WATT-HR	BMS & Middleware
Heat Meter	ThermalStore HM_ENERGY WATT-HR	BMS & Middleware
Gas Meter	Gas Meter Data	BMS & Middleware
Gas Meter	Boiler Gas	BMS & Middleware
Gas Meter	CHP Gas	BMS & Middleware
Electricity Meter	Electricity Meter Data	BMS & Middleware
Electricity Meter	Main ESB (50001)	BMS & Middleware

Electricity Meter	50001.AI4	BMS & Middleware
Electricity Meter	50001.AI5	BMS & Middleware
Electricity Meter	50001.AI6	BMS & Middleware
Electricity Meter	50001.AI7	BMS & Middleware
CHP Actuation		BMS & Middleware
Heat Pump Actuation		BMS & Middleware
EV Charger Actuation	cgi-zappi-mode	Gateway & Middleware
EV Electricity Meter	cgi-jdayhour-Z15903784/genpos	Gateway & Middleware
Solar Electricity Meter	cgi-jdayhour-Z15903784/gep	Gateway & Middleware
Water Heater Actuation	cgi-eddi-boost- Z15903784/h1d	Gateway & Middleware
Water Heater Electricity Meter	cgi-cgi-jday- Z15903784/h1d	& Middleware
Water Heater Temperature	cgi-jday- Z15903784/pt1	Gateway & Middleware

In order to validate the data and to establish a baseline, ARDEN obtained data from the on-site BMS dating back to January 2019 for each sensor and device on the system. The data is in five-minute interval form compiled in separate CSVs depending on whether it is electricity or gas consumption or air quality data. The data was validated by site visits where meter readings were taken for each meter. After the second visit the readings were compared to data collected from the BMS over the same time period. The historical data was then used to generate graphs on average daily, monthly and yearly consumption as well as efficiency of devices, air quality profiles and demand profiles. Some of these are displayed below. ARDEN relayed this information back to the Rediscovery Centre and advised them on the inefficiencies uncovered.

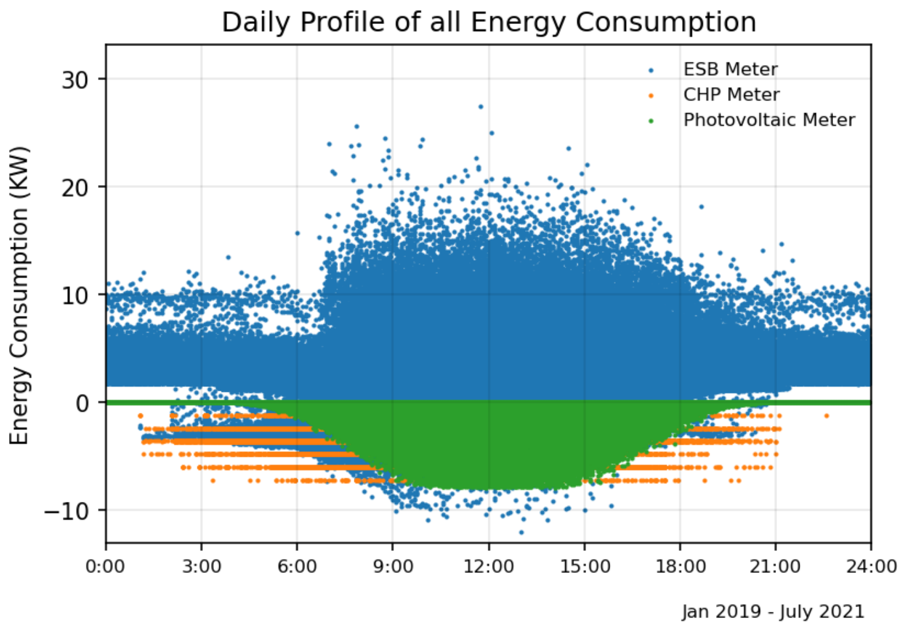


Figure 14 Average daily profile of energy consumption

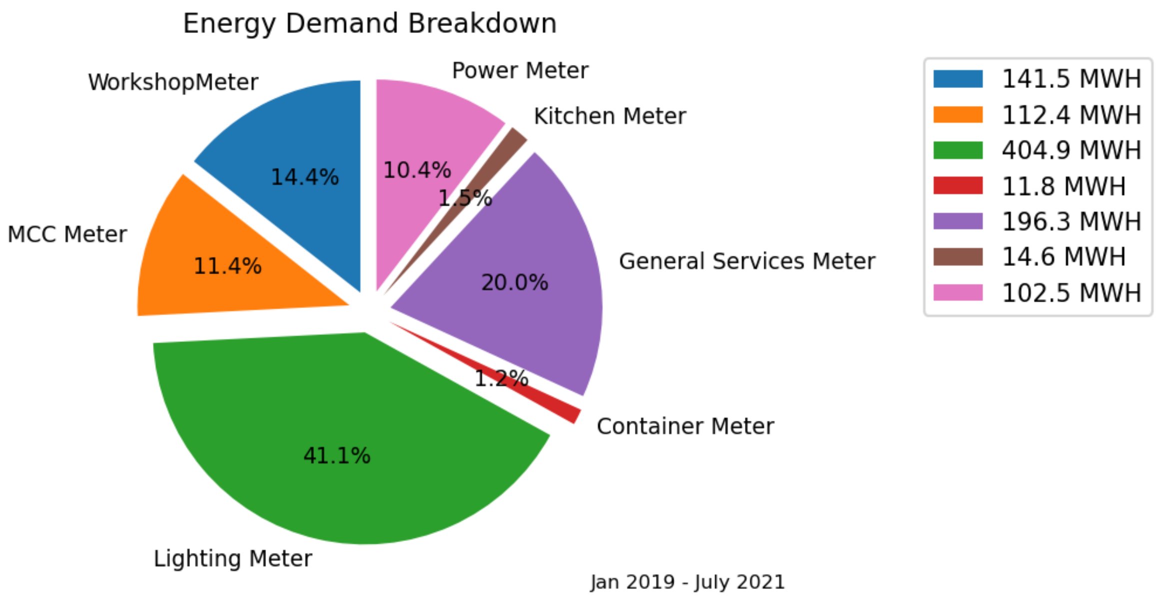


Figure 15 Energy demand at commercial site

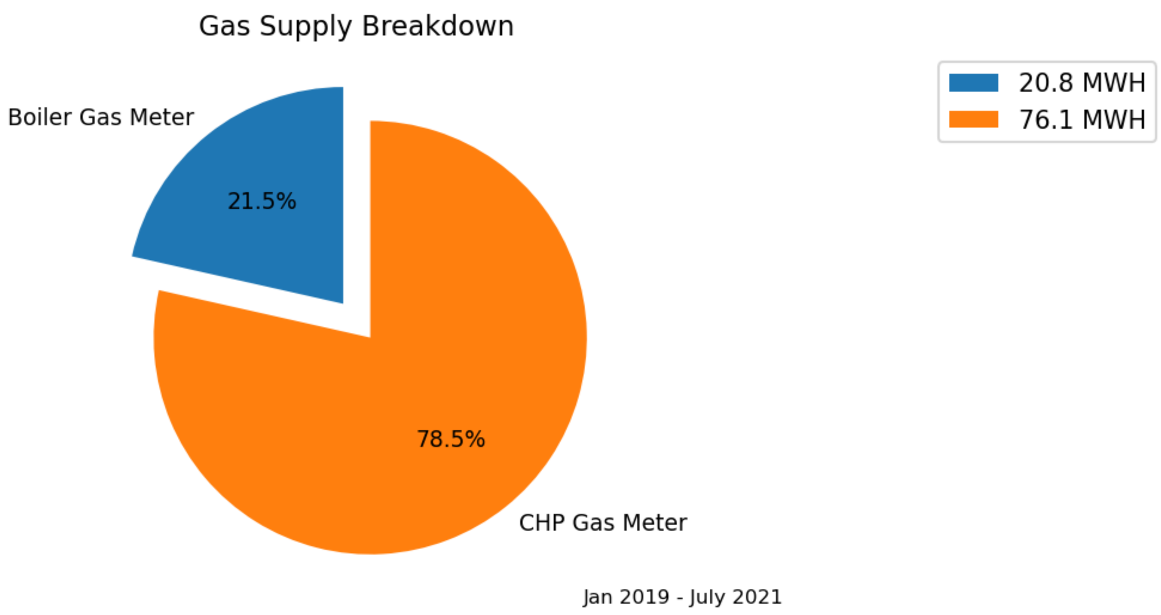


Figure 16. Gas supply at commercial site

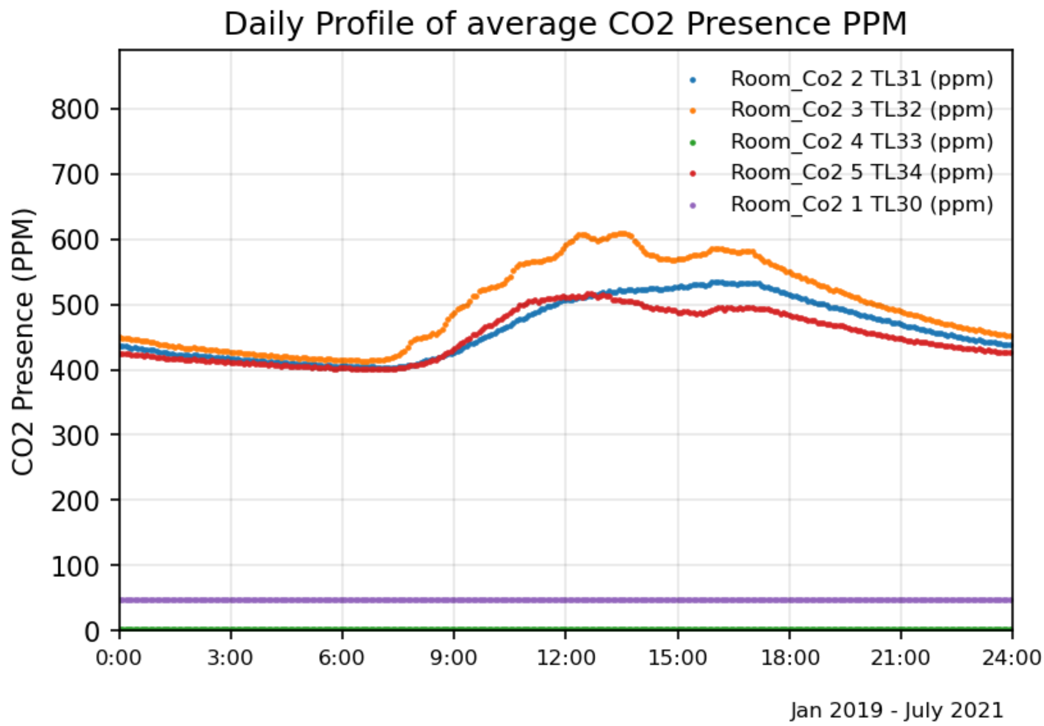
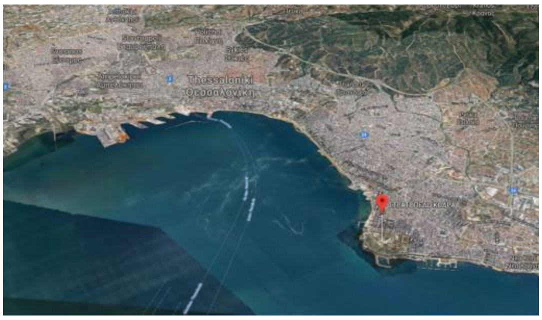


Figure 17. Average daily profile of CO_2 presence

6. KAMA Pilot

6.1. Description

In Greece, the Greek Army usually offers to its staff housing facilities. These facilities belong to the main body of the Greek Army and they are offered to a selected number of executives and lower level staff as an extra benefit other than their normal salaries. Most of Greek Army facilities have been constructed in the distant past and have not been maintained in optimum condition. However, during the last years the general recession increased the importance of the housing benefit offered by the body of the Greek Army to its staff. Therefore, there is an increasing number of renovations taking place in such premises. One such example is a small group of houses located in Dalipi area in Thessaloniki.



6.2. Use Cases

The use cases that describe the Greek pilot's functionalities supported by the PHOENIX framework are the following:

<u>UC01: Adapt & Play integration of domestic appliances, legacy equipment and building systems</u>

This use case relates to the smart hardware that is installed in the Greek pilot to increase the level of smartness of the building.

<u>UC03: Services for building occupants to maximize their energy efficiency and increase overall building performance</u>

This use case deals with occupants' engagement in actions that allow monitoring buildings' energy performance and promote energy savings.

UC04: Provision of Comfort, Convenience and Wellbeing services to building occupants

This use case concerns services that aim to improve life quality via simplifying everyday activities.

UC05: Portfolio flexibility analysis and configuration to optimize grid operation

Grid flexibility, by the introduction of power electronics and storage units, is under investigation, since Greek energy model is static and currently there is no framework in place that embraces the provision of flexibility services. The implementation of this use case could potentially be done in a virtual environment.

UC06: Flexible billing services and smart contracts for the retailer customers

Greek energy retail market is based on static pricing; therefore this use case could potentially be performed by a simulated concept where retailers react according to energy demands and rates of the energy exchange.

6.3. Legacy equipment and systems

Originally the Greek pilot building was equipped with the basic infrastructure for electrical heating, cooling and conventional lighting, with no sophisticated functions that could consider the individual flat needs for heating/cooling. Moreover, there was no energy generation from photovoltaics nor any solar thermal system that could cover the domestic hot water (DHW) demands. During the renovation process, and before PHOENIX project initiation, several pieces of hardware were installed for all the main domains, which comprise the legacy equipment, and they include the following:

- Eight independent heat pumps, one for each apartment, controlled by simple, conventional thermostats, for the heating and cooling of the apartment.
- LED lights at all the apartments and the common areas.
- Eight solar thermal boilers with simple differential thermostats, one for each apartment, for domestic hot water production.
- Photovoltaic panels and inverters of 19,8 kW at the roof of the building for solar energy production.
- A battery of 5.1 kWh for energy storage.
- An Electric vehicle charger for residents’ vehicles charging.
- Eight manual shading systems, one at each flat’s front view.

The basic characteristics and specifications of all the legacy equipment are summarised in Table 7. However there exists no computer-based system to control or monitor the building's mechanical and electrical equipment. There is only the option of controlling the room temperature via the thermostat and the possibility of having an overview of the solar energy production via the inverters' manufacturer platform (Fronius solar web).

Table 7 Legacy equipment of the Greek pilot

Domain	**Details of equipment (e.g. capacity, model, power etc)**
Heating & Cooling system	8 pcs Carrier Fancoil 42N, 8 pcs Hitachi Yutaki-M 4NE
Led- Lighting	Various types of led lamps and bulbs
Domestic Hot Water	8 pcs solar thermal boilers, 8 pcs Soller ST112 differential thermostat
Renewable energy	60pcs of Canadian Solar 330Wp, 1pc Fronius Symo 20, 1pc Fronius Gen24 6kW
Energy storage	1 pc BYD HVS 5,1 kWh
EV charger	1 pc Noark EV wall-mounted charger Ex9EV3 T2 10A
Dynamic Envelope	8 pcs 2m long 1,5 m wide custom made shading system

6.4. Interventions and Requirements

The necessary interventions for the Greek pilot have to do with the controlling and monitoring of the legacy equipment. More specifically, several sensors were installed to achieve the Key Performance Indicators (KPIs) set in DoA for all the domains, and meanwhile increase the energy performance and the SRI of the residential building. The information that is needed from the monitoring system is concerned with energy services, namely energy production and consumption, energy and cost savings from the consumed electrical power and on the other hand with non-energy services, namely the quality of life of the residents. To reach these goals, the residents shall get relevant notifications about everyday activities from the platform and act accordingly. The notifications will be the outcome of the algorithm, which will take into consideration the preferences of the users, relative readings from the sensing devices and weather data from external data sources (i.e. Weatherbit).

The sensing devices that need to be installed to the Greek pilot are summarised in Table 8. The readings mainly include electrical consumption and generation, temperature (for the indoor environment and the domestic hot water), luminance, and CO_2 concentration levels. In addition, there will be installed one smart plug and one smart LED lamp in each apartment to increase the comfort levels of the residents.

There are two types of sensors as regards their connectivity/ communication protocols: Modbus RTU and Z-wave. All these devices will be connected to a proper gateway (Raspberry pi), as designed and proposed by the pilot manager. The Modbus sensors will pass through a Modeth module before the end up to the Raspberry pi via Ethernet connection, while the Z-wave sensors are directly communicating with a USB Z-stick, which is plugged in the Raspberry pi.

Table 8. Monitoring devices of the Greek pilot

Domain	Details of sensors (e.g. model, type, communication protocol etc)
Heating & Cooling system	8 pcs MCO Home 4 MH8-FC4 (Z-wave)
Led- Lighting	8 pcs Aeotec LED Bulb 6 Multi White (Z-wave)
Domestic Hot Water	3 pcs FIBARO Smart Implant (Z-wave), 16 pcs DS18B20 temperature sensors
Energy	8 pcs single phase smart meters Orno WE-504 (Modbus), 16 pcs single phase smart meters Orno WE-514 (Modbus), 8 pcs three phase smart meters Orno WE-516 (Modbus), 1 pc Fronius three phase smart meter TS 50kA-3 (Modbus), 8 pcs Smart switch 7 (Zwave)
EV charger	1 pc single phase smart meter Orno WE-514 (Modbus)
Dynamic Envelope	8 pcs Aeotec Nano Switch, 1 pc Aeotec Multisensor 6
Air quality	MCO Home MH9-CO2 (Z-wave)
Gateways	2 pcs Ampero mod-eth modules, 1 pc Raspberry pi 3 Model B+, 1 pc Aeotec Z-Stick Gen 5

Hence, the interventions that are required for the Greek pilot include solely the installation of sensors, IoT devices and gateways. One issue to be considered is the communication of the wireless devices (Z-wave) with the gateway, due to the relatively large distances from one flat to the other and to the rooftop, where the gateway is placed. In case that such issues arise, the installation of a second gateway at an appropriate location is recommended to reinforce the strength of the signals.

6.5. Systems

As mentioned before, there is no BMS or any other computer-based platform installed in the Greek pilot to act as a monitoring tool. The monitoring and notification platform will come through the PHOENIX dashboard.

7. MIWENERGIA

7.1. Description

MIWenergia´s pilot site is located at Region de Murcia and includes two different types of buildings, one commercial and one residential. The demonstration activities for the commercial building will take place at CEEIC in Cartagena. This business incubator building focuses on start-up and early-stage innovative companies. 20 spaces between company's offices and lecture rooms will participate in PHOENIX project.

Figure 19. Commercial pilot, CEEIC Murcia

Figure 20. Residential pilot Murcia

The residential building selected is located in the city centre of Murcia. Four apartments will be involved in the pilot site. Each apartment is approximately 125 square meters and they are equipped with common domestic appliances.

7.2. Use Cases

The use cases for the Spanish pilots site supported by the PHOENIX framework are the following:

UC01: Adapt & Play integration of domestic appliances, legacy equipment and building systems

This use case will be implemented in all pilot sites through the deployment of smart meters and smart devices and integration in the PHOENIX solution.

UC02: Building knowledge enhancement to upgrade the smartness of buildings

PHOENIX solution will provide the availability of consumption and comfort data as well as enhanced control of HVAC units in the office building.

UC03: Services for building occupants to maximize their energy efficiency and increase overall building performance

The integration of the smart devices will give information to the users and will allow monitoring energy efficiency and promote energy service and more efficient energy behaviour.

UC05: Portfolio flexibility analysis and configuration to optimize grid operation

Spanish Pilot sites will be ready to participate in demand response actions and use their flexibility for grid optimisation thanks to the PHOENIX framework.

UC06: Flexible billing services and smart contracts for the retailer customers

Residential users have changed their tariffs to dynamic tariff in which the price depends on the wholesale market hourly prices. Load shifting will automatically translate into energy savings for the users. In the office building, due to the existing time of use tariff, this savings can be also achieved thanks to the control of HVAC units that could allow pre-heating or pre-cooling in order to minimise consumption during peak hours.

7.3. Legacy equipment and systems

The existing equipment of this pilot site is summarized in Table 9.

Table 9. Legacy equipment for the Spanish demo site

Domain	**Details of equipment (e.g. capacity, model, power etc)**	**Use Cases related**
Commercial building		
Heating & Cooling system	Central HVAC system for the common areas in the office building that operates without central control system. Individual HVAC systems for each office.	UC01, UC02, UC03, UC05, UC06
Lighting	Lighting system with individual switches per office and fluorescent technology	UC2
Domestic Hot Water	Electric DHW tank	N/A
Monitoring and control	Smart meter that provides hourly consumption data at building level. Building manager receives information from the retailer with the monthly cumulated consumption in their electricity bill.	UC01, UC02, UC03, UC05, UC06
Residential buildings		
Heating & Cooling system	Duct heat pumps	UC01, UC02, UC03, UC06
Lighting	Lighting system with individual switched per room and LED & fluorescent technology.	UC2
Domestic Hot Water	Gas boiler	N/A
Monitoring and control	Smart meter for every apartment that provides hourly data related to energy demand	UC01, UC02, UC03, UC05, UC06

7.4. Interventions and Requirements

The state of the buildings before the start of PHOENIX project was practically non-smart, with no BMS nor any control device sending any type of information to the users, which sets the preliminary SRI overall values under 9%. Hence, several devices and sensors need to be installed in order to be able to fulfil the Key Performance Indicators (KPIs) set in the DoA. There are two types of sensors as regard as their connectivity/communication protocols. Z-Wave and WMP/ASCII. The list of devices is shown in Table 10.

Table 10 Devices installed in Spanish demo site

Domain	Details of equipment (e.g. capacity, model, power etc)	Use Cases
Monitoring and control	**Monitoring devices (electrical consumption):** three phase & one phase smart meters. ZWave. • Qubino 3-Phase Smart Meter • Aeotec Home Energy Meter - three clamps GEN5 • Zipato Energy Meter – 1 phase (2 circuits) • WiDom Energy Driven Switch C version	UC01, UC02, UC03, UC05, UC06
Heating & Cooling system	**Monitoring devices (temperature & Humidity):** temperature and Humidity ZWave sensors - MCOHome MH9	UC01, UC02
Ventilation	**Monitoring devices (air quality):** CO_2 Zwave sensors - MCOHome MH9	UC01, UC02
Monitoring and control	**Gateway:** Raspberry Pi 3 Model B+ with Z-Wave Aeotec Z-Stick Gen5 USB.	UC01, UC02, UC03, UC05
Heating & Cooling system	**HVAC smarts control devices:** Intesis Box WMP – Universal IR	UC01, UC02, UC03, UC05, UC06

Smart meters will be installed in the main electric boards of each dwelling and in the ones that control each office of the commercial building. These devices will measure the total consumption and the HVAC consumption. Moreover, a sensor that measures temperature, humidity and CO_2 will monitor the comfort parameters and air quality in every zone. HVAC smart devices will be deployed to control the split units existing in the offices.

All these devices will be connected to a gateway (Raspberry pi), one per dwelling and twelve in the office building. The Z-Wave sensors communicate with the gateway with an USB Z-stick plugged in the Raspberry pi. The HVAC devices, which communicate via Wi-Fi, will be integrated to the gateway by ODINS.

After the installation of these devices, it is estimated that significant improvement in the SRI of the buildings and an increase in the energy efficiency will be provided. Users will be able to have constant information regarding their consumption, will receive notifications and suggestion to increase efficiency and reduce costs (implicit demand response actions). HVAC control devices will allow offices to interact with the grid though explicit demand response actions. Comfort and air quality sensors will give the users tools to improve their health and wellbeing.

7.5. Systems

The monitoring and notification platform will come through the PHOENIX dashboard that will get all data from the gateways and will show the information and will provide notifications of possible actions to increase energy efficiency, comfort and to participate in the electric market.

8. LTU

8.1. Description

The LTU pilot site is located in Skellefteå, a city in northern Sweden. The climate conditions in this zone are characterized by a cold winter with snow and a mild summer. The pilot site includes a building which is both residential and commercial. It has 12 apartments and a commercial space at the front of the building on the ground floor. The total area including the commercial space is 1920 square meters of heated area and 1278 square meters of living area. The building has big apartments with 4-5 rooms each. In addition to the apartments there is a community laundry and a common room for socializing. There are 4 outdoor parking places on the street with electric points for heating the cars and 10 indoor garages for cars. The building was built in 1966 and is made of concrete with 3 pane glass windows. It has an FX system for ventilation with two fans for supply and exhaust air with a rotating heat exchanger where 80% of the energy is sent back into the building. The occupant age group ranges between 20-76 years with average age of approximately 50 years. All apartments are equipped with appliances owned by the owners.

Figure 21: Skellefteå Pilot Building.

8.2. Use Cases

The Swedish pilot site addresses the following three objectives:

Objective 1 (O01): Energy savings identified as follows: 1) Optimization of heating systems in for the building and 2) Investigating if energy efficiency in lighting systems and car heaters are possible to improve.

Objective 2 (O02): Comfort and Convenience: the use of sensors for e.g. internal environmental conditions can improve ambient conditions for building and facilitate different systems operational control. This objective is concerned with aspects of people's everyday living. The occupants of the building might feel more comfortable in terms of temperature and air quality conditions due to sensor-based improved temperature control. Sensors are installed in common areas, and in some selected apartments based on consent from the occupants.

Objective 3 (O03): Improvement of the cost of energy: This objective is concerned with the optimization of energy consumption. This will be achieved by improving the regulation of the district heating-based heating system and installing smart thermostats.

The use cases for the Swedish pilot site supported by the PHOENIX framework are the following:

UC01: Adapt & Play integration of domestic appliances, legacy equipment and building systems

This use case relates to the smart hardware that is installed in the Swedish pilot to increase the level of smartness of the building. The related actors in this use case are building occupants and building managers.

UC02: Building knowledge enhancement to upgrade the smartness of buildings

This use case also relates to the smart hardware that is installed in the Swedish pilot to increase the level of smartness of the building. The related actors in this use case are building occupants and building managers.

UC03: Services for building occupants to maximize their energy efficiency and increase overall building performance

This use case deals with occupants' engagement in actions that allow monitoring buildings' energy performance and promote energy savings. The related actors in this use case are building occupants.

UC04: Provision of comfort, convenience and well-being services to building occupants

This use case concerns services that aim to improve quality of life via simplifying everyday activities. The related actors in this use case are building occupants.

8.3. Legacy Equipment and Systems

The details of the legacy equipment and systems and the links to the objectives and use cases for the Swedish pilot site is shown in Table 11.

Table 11 – Legacy Devices at the Swedish Pilot

Details of equipment (e.g. capacity, model, power etc)	Pre-existing	Objectives	Use Cases
Heating system: Heat exchanger on the secondary side of the heating system is a Parca Norrahammar, type RL-01 from 1982	YES	O01, O02, O03	UC1, UC2, UC3, UC4
Ventilation system: FDX system for building ventilation	YES	O01, O02, O03	UC1, UC2, UC3, UC4
DHW: Linked to the heating system	YES	O01, O02, O03	UC1, UC2, UC3, UC4

8.4. Interventions and Requirements

Table 12 shows the details of the new equipment and integrations as well their links to the use cases.

Table 12 – New Devices and integration at the Swedish Pilot

Details of Equipment	Pre-existing	Objectives	Use cases
Communication: KTC IMC equipment for communication with KTC 1235 modem	NO, installed in the building and integrated with new system	O01, O02, O03	UC1, UC2, UC3, UC4
Monitoring devices (temperature): building's external temperature, apartments' temperature, thermostats	Building's external temperature: YES Apartment's temperature and thermostats: NO	O01, O02, O03	UC1, UC2, UC3, UC4
Monitoring devices (air quality): CO_2 sensor	NO	O02	UC4
Gateway: Raspberry Pi, Z-Wave USB	NO	O01, O02, O03	UC1, UC2, UC3, UC4

The legacy equipment and system are integrated to communicate with the LTU FIWARE platform and the Phoenix platform. These integrations can be divided into three parts. First is the installation of a new device from KTC called 'IMC' which is installed so the legacy modem KTC1235 can be communicated with the newer KTC systems to send data outside of the legacy system. Along with the IMC box, an API is built and provided to send data via JSON to the LTU FIWARE platform. Second, an API is built to provide system-to-system integration between LTU FIWARE platform and PHOENIX platform. This is now tested and operational. This will enable communication of all HVAC data from the legacy system to the PHOENIX platform.

Lastly, is the integration and installation of thermostats and air quality sensors using the Raspberry Pi gateway. This is for the individual apartments in the Skellefteå pilot site where data is collected in the Raspberry Pi gateway and then sent to the PHOENIX platform using node-red integration provided by ODINS and UMU. This is initially implemented for one of the apartments and tested by LTU. This will be followed by more apartments who show interest in the same building for a similar setup. The XML for configuration and provisioning of Z-Wave sensors using a Raspberry Pi as gateway has been configured. The final installation and testing is underway and will be finalized in the coming weeks. The setup follows the same procedures as outlined in their pilot for the Z-wave devices.

8.5. Systems

The monitoring and notification will be performed using the PHOENIX platform. There is currently no system in place for this at the Skellefteå pilot site. Some of the notifications may also go through the LTU FIWARE platform.

9. Integration of Legacy Devices

The communication with legacy devices (and Z-Wave devices) is managed in most of the cases by Raspberry Pi gateways. In the UMU pilot, where ODINS gateways are installed, the architecture used is also based on a Raspberry Pi, but it contains an additional board that provides some extra features.

This image is designed to be modular and, with the proper configuration, can be used to control devices from multiple technologies at the same time, including Modbus-RTU or Modbus/TCP, wired sensors and actuators (ODINS gateway only), Z-Wave and Intesis devices compatible with the WMP protocol. These last ones are universal IR transmitters/receivers that control splits and use a proprietary (but documented) protocol.

For Modbus devices and for wired sensors/actuators, the communication between the image and the platform goes through a MQTT-JSON IoT-Agent while for Z-Wave and Intesis WMP devices, the information goes directly to the Context Broker for readings and writings (writings use an agent that is also documented in this deliverable in a following section).

9.1. Modbus

Modbus is a standard protocol for communicating with devices that uses a master ↔ multislave model where each slave has a unique address (1 byte). It's widely used in industrial environments to control Adjustable Frequency Drives (AFDs), read power meters, etc.

The most common variants are Modbus-RTU, where the physical layer is a RS-485 bus, and Modbus/TCP, which uses TCP/IP networks. The main difference between them is that Modbus/TCP adds an extra header to encapsulate the data exchanged in TCP connections, but other than that, the requests and responses use the same format as Modbus-RTU. In addition it allows multiple masters to have access to a single bus because each Modbus/TCP gateway can keep multiple TCP connections active at the same time. This gateway acts as a single master from the bus perspective, which means only one transaction can be active in the bus and multiple requests must be serialized if more are received while there's one in progress. In the end the master ↔ multislave restriction must be respected in the bus side.

From a global perspective, Modbus represents the data as an array of registers (16-bit per register) or discretes/coils (digital 1-bit inputs/outputs) independent of what's being stored in each entry (register, discrete or coil).

In order to access to the data, different functions are offered to provide read and write support. For PHOENIX, only register mode access is necessary and, in terms of Modbus functions, it means only the next functions are implemented:

- Read Holding Registers (0x03).
- Read Input Registers (0x04).
- Write Single Register (0x06).
- Write Multiple Registers (0x10).

A sample exchange to get the first Holding Register of slave with address 1 would have the next format (Modbus-RTU, with the value of this register being 0x1234):

Request							
Slave address	Function code	Register offset (2 bytes)		Number of registers (2 bytes)		Checksum (2 bytes)	
0x01	0x03	0x00	0x00	0x00	0x01	0x84	0x0A

Response						
Slave address	Function code	Number of bytes received	Value of register (with 16-bit registers, 1 register = **2 bytes**)		Checksum (2 bytes)	
0x01	0x03	**0x02**	0x12	0x34	0xB5	0x33

Based on this, the solution has been divided in three abstraction levels in order to make it as generic as possible.

9.1.1. Low abstraction level

This level has to do with the implementation of the Modbus functions based on the format described by the standard, and includes concepts such as slave address, function code, start register offset, etc. It could be seen as a sort of *Communication Layer* that has some differences in the physical part depending on whether we're dealing with Modbus-RTU (RS-485 bus) or Modbus/TCP.

9.1.2. Mid abstraction level

In this level 5 concepts have been defined:

- Modbus/TCP node/gateway (only for Modbus/TCP), identified by a pair IP/hostname + port.

- Modbus device, identified by its slave address and optionally by the Modbus/TCP node that provides access to the device (only Modbus/TCP).
- Modbus read area (one or more per Modbus device), defined by the function code to use (Read Holding Registers or Read Multiple Registers), the offset of the first register and the number of registers to read.
- Modbus read variable (one or more per Modbus device). Here is when the data decoding is performed as it extracts values from registers based on the offset of the register, the data type (signed or unsigned 16/32-bit integer, float, etc.), a multiplier, period for generating historic trends, description of the attribute used for sending the value inside the json payload of the MQTT publication, etc.
- Modbus write variable (one or more per Modbus device). In this case the operation is performed using the standard command mechanism of the PHOENIX MQTT IoT-Agent. The definition of the variable is similar to the previous case.

These two abstraction levels are implemented in the service that is running in the gateway image and guaranties the compatibility with almost every Modbus device. A set of command-line tools is available to configure the Modbus interface so that the service can read/write Modbus variables, which is the key concept here. These tools interact with the service through D-Bus [4], a standard inter-process communication mechanism widely used in Linux systems.
A sample script that configures/enables a Modbus-RTU device is the next one:

```
#!/bin/bash

if [ $# -ne 4 ]; then

	echo "First parameter = Modbus device number (from 1 to 32), second parameter = Description, third parameter = Modbus slave address and fourth parameter = seconds between reading rounds"

	exit 0
fi

dbus-send --system --print-reply --dest=es.odins.iotconnector.AppServer "/es/odins/iotconnector/AppServer" "es.odins.iotconnector.AppServer.ModbusReadDevicesConfigureDevice" byte:$1 string:"$2" byte:$3 uint16:$4
```

9.1.3. High abstraction level

After having defined the previous levels, several scripts that use the command-line tools mentioned in the previous section have been created to simplify the configuration of well-known devices.

As stated regarding the Modbus protocol, it only offers the information in a generic way. Each manufacturer provides the register map to let integrators know which information is available, where to find it and which format it has (register offsets, multipliers, etc.). Based on this, scripts for all the devices of the project have been created so that the configurations are generated only requiring a few general parameters.

A sample script for a single-phase Modbus/TCP power meter is the next one (these few general parameters required have been highlighted in bold at the beginning of the script):

```
#!/bin/bash

if [ $# -ne 6 ]; then
	echo "First parameter = Modbus/TCP node number (from 1 to 8), second parameter = Modbus device number (from 1 to 32), third parameter = Modbus slave address, fourth parameter = Description, fifth parameter = minutes between power readings (5, 15, 30, 60) and sixth parameter = minutes between energy readings (5, 15, 30, 60)"

	exit
fi

# this value will be fixed for now
SECONDS_BETWEEN_READINGS=30

# references to the scripts required for configuring devices, areas and variables
CONF_DEVICE="dlmTcpConfDispositivo"
CONF_AREA="dlmTcpConfArea"
CONF_VARIABLE="dlmTcpConfVariable"

# we will use an already-existing node (must be created previously!)
NUM_MODBUS_TCP_NODE=$1

NUM_MODBUS_DEVICE=$2
MODBUS_SLAVE_ADDRESS=$3
DESCRIPTION="$4"

# create the device
"${CONF_DEVICE}" ${NUM_MODBUS_TCP_NODE} ${NUM_MODBUS_DEVICE} "${DESCRIPTION}" ${MODBUS_SLAVE_ADDRESS} ${SECONDS_BETWEEN_READINGS}

#### add the 2 areas
OFFSET_ACTIVE_POWER=320
OFFSET_ACTIVE_ENERGY=40962

# active power = offset 320 (LONG/INT32)
USE_READ_INPUT_REGISTERS="false"
OFFSET=${OFFSET_ACTIVE_POWER}
```

```
NUMBER_OF_REGISTERS=2

"${CONF_AREA}" ${NUM_MODBUS_TCP_NODE} ${NUM_MODBUS_DEVICE} 1 "${USE_READ_INPUT_REGISTERS}" $OFFSET ${NUMBER_OF_REGISTERS}

# active energy = offset 40962 (ULONG/UINT32)
USE_READ_INPUT_REGISTERS="false"
OFFSET=${OFFSET_ACTIVE_ENERGY}
NUMBER_OF_REGISTERS=2

"${CONF_AREA}" ${NUM_MODBUS_TCP_NODE} ${NUM_MODBUS_DEVICE} 2 "${USE_READ_INPUT_REGISTERS}" $OFFSET ${NUMBER_OF_REGISTERS}

#### add the 2 variables
# data types -> (1=UINT16, 2=UINT32, 3=INT16, 4=INT32, 5=FLOAT32, 6=DOUBLE64, 7=BCD16, 8=BCD32)

# active power = offset 320 (LONG/INT32)
USE_READ_INPUT_REGISTERS="false"
OFFSET=${OFFSET_ACTIVE_POWER}
DATA_TYPE=4
MULTIPLIER="0.001"
NUM_DECIMALS=3
MINUTES_BETWEEN_POWER_READINGS=$5

"${CONF_VARIABLE}" ${NUM_MODBUS_TCP_NODE} ${NUM_MODBUS_DEVICE} 1 "Active Power" true "${NUM_MODBUS_TCP_NODE}-${NUM_MODBUS_DEVICE}-totalActivePower" $OFFSET ${DATA_TYPE} "${MULTIPLIER}" "${USE_READ_INPUT_REGISTERS}" ${NUM_DECIMALS} ${MINUTES_BETWEEN_POWER_READINGS}

# active energy = offset 40962 (ULONG/UINT32)
USE_READ_INPUT_REGISTERS="false"
OFFSET=${OFFSET_ACTIVE_ENERGY}
DATA_TYPE=2
MULTIPLIER="0.01"
NUM_DECIMALS=2
MINUTES_BETWEEN_ENERGY_READINGS=$6

"${CONF_VARIABLE}" ${NUM_MODBUS_TCP_NODE} ${NUM_MODBUS_DEVICE} 2 "Active Energy" true "${NUM_MODBUS_TCP_NODE}-${NUM_MODBUS_DEVICE}-totalActiveEnergyImport" $OFFSET ${DATA_TYPE} "${MULTIPLIER}" "${USE_READ_INPUT_REGISTERS}" ${NUM_DECIMALS} ${MINUTES_BETWEEN_ENERGY_READINGS}
```

9.2. Wired sensors and actuators

This functionality is only available in ODINS gateways as it requires the external board designed by ODINS that adds direct support to some wired input/outputs as well as a CAN BUS interface [5]. Using the CAN BUS interface, more slave modules can be connected to increase the number of inputs/outputs supported with CANopen being the communication protocol used between the gateway and the slaves. This external board is plugged in the expansion connector of the Raspberry Pi.

In this case there is an additional service that offers a transparent access to the inputs/outputs regardless of whether they are in the gateway itself or in any of the slaves. This low-level service is fully integrated with the main service (the one used for Modbus devices) and, with both working together, the interaction between the platform and the sensors/actuators also goes through the PHOENIX MQTT IoT-Agent.

9.3. Z-Wave

As stated in D3.1, Z-Wave is a standard wireless protocol mostly used in home automation capable of generating meshed networks in a way that each node can act as a router.

From an integration perspective in the context of the project, Z-Wave devices are still being controlled by a set of Node-Red flows, but they've been modified in comparison to the preliminary version used in the PoC.

As with Modbus devices, the solution for Z-Wave has been divided in three abstraction levels to make it as generic as possible.

9.3.1. Low abstraction level

This level has to do with the communication itself, which is controlled by the *Open-Z-Wave* library exactly as it was in the previous set of flows.

At this point it's important to understand how certain Z-Wave devices work. Some of them publish readings if there's a change big enough in comparison to the last value sent. Normally (but not always) the threshold that determines if the value must be sent can be configured.

However others don't publish data on their own no matter what happens. The only way to force this communication is by sending the *refreshValue* command, which asks the device to send the current value towards the concentrator. In fact there are even devices that somehow combine both criteria when it comes to publish data, which means even if the *refreshValue* command is sent, the data won't be published unless the change is big enough.

On the other hand, actuation over devices is done using the *setValue* command.

In both cases the information is grouped around the concept of *Command Classes*. Every class offers certain functionality, and each specific attribute is identified by 4 parameters:

- *Node id*. That's the identifier assigned to the device when it's paired with the USB stick.

- *Command class*. Each class has a unique well-known identifier.
- *Instance*. A device can provide 1 or more readings of the same type and each reading could be seen as a subset of the class (that's what each instance is representing).
- *Index*. For each class, well-known indexes are assigned to different types of readings.

9.3.2. Mid abstraction level

In this level the configuration information can be extracted directly from the .xml file exported by the *Open-Z-Wave* library when the *writeConfig* command is executed. As a result the configuration file is stored in the Node-Red base folder (usually in *$HOME/.node-red*) with a content similar to the next one (compact version):


```
<?xml version="1.0" encoding="utf-8" ?>
<Driver xmlns="http://code.google.com/p/open-zwave/" version="3" home_id="0xc2aa08bb" node_id="1">
	<Node id="1" name="" location="" basic="2" generic="2" specific="1" type="Static PC Controller" >
		<Manufacturer id="86" name="Aeotec">
			<Product type="1" id="5a" name="Z-Stick Gen5" />
		</Manufacturer>
	</Node>
	<Node id="2" name="" location="" basic="4" generic="33" specific="1" roletype="5" devicetype="3328">
		<Manufacturer id="15f" name="Mcohome">
			<Product type="901" id="5102" name="Unknown: type=0901, id=5102" />
		</Manufacturer>
		<CommandClasses>
			<CommandClass id="49" name="COMMAND_CLASS_SENSOR_MULTILEVEL" version="7" innif="true">
				<Instance index="1" />
				<Value type="decimal" genre="user" instance="1" index="1" label="Temperature" units="C" />
				<Value type="decimal" genre="user" instance="1" index="5" label="Relative Humidity" units=""
/>
				<Value type="decimal" genre="user" instance="1" index="17" label="CO2 Level" units="ppm" />
			</CommandClass>
		</CommandClasses>
	</Node>
	<Node id="3" name="" location="" basic="4" generic="49" specific="1" roletype="5" devicetype="4096">
		<Manufacturer id="86" name="Aeotec">
			<Product type="2" id="5f" name="Unknown: type=0002, id=005f" />
		</Manufacturer>
		<CommandClasses>
			<CommandClass id="50" name="COMMAND_CLASS_METER" version="4" request_flags="3" innif="true">
				<Instance index="1" />
				<Instance index="2" endpoint="1" />
				<Value type="decimal" genre="user" instance="1" index="0" label="Energy" units="kWh" />
				<Value type="decimal" genre="user" instance="1" index="8" label="Power" units="W" />
				<Value type="decimal" genre="user" instance="2" index="0" label="Energy" units="kWh" />
```

```
                                                  <Value type="decimal" genre="user" instance="2" index="8" label="Power" units="W" />
                                        </CommandClass>
                              </CommandClasses>
                    </Node>
</Driver>
```

In the previous example the network is formed by 3 nodes:

- Node 1. It's an identifier reserved for the USB stick itself as it is represented in the configuration file as another member of the network.
- Node 2. This multisensor offers temperature, humidity and CO_2, and the four parameters that identify each single reading are highlighted in bold (node id, command class, instance and index).
- Node 3. This is a power meter that provides values from multiple phases, each one of them being stored in a separate instance of the same command class.

Here the information is extracted manually from the .xml file going directly to the section of the right class (depending on the functionality provided by the device). It's important to match each node id with the right device, and to do that, a first Node-Red flow was created.

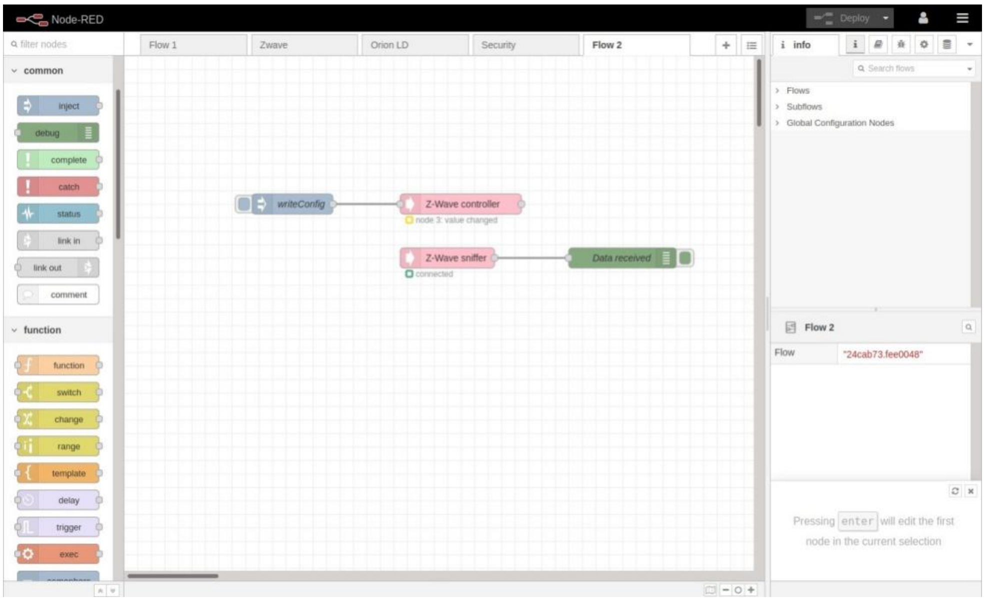


Figure 21. Z-Wave / Config extraction flow

Following a strict protocol based on pairing a device with the USB stick, waiting for it to communicate, forcing a *writeConfig* operation and extracting its node id from the .xml file (a new node entry will appear in the file every time a device is paired), all the devices can be safely identified. This is a critical step as, for many devices, there's no easy way to access to their identifiers and in case multiple devices of the same kind were installed, some of them could get mixed in the identification process.

This process can be simplified by pairing the devices with the USB stick in a well-known order, waiting for all of them to communicate and then forcing a *writeConfig* operation. By default the USB stick will increase the node identifier assigned every time a new device is paired but this might not work exactly this way and that's why the step-by-step alternative is safer and is the preferred one even though it's slower.

9.3.3. High abstraction level

In this final level some scripts have been created to get the functionality offered by a whole Z-Wave network based on the set of identifiers obtained in the previous step and a list of well-known manufacturers and models including those that have been installed in the project. Either processing the .xml file manually or automatically with these scripts, the information needed to configure the flows and later the entities can be extracted.

An example of the output of the script used on the previous .xml file is the next one:

[2,49,1,1] MCO Home CO2 / Temperature
[2,49,1,5] MCO Home CO2 / Humidity
[2,49,1,17] MCO Home CO2 / CO2
[3,50,1,0] Aeotec 3-phase Meter / Phase 1 / Energy
[3,50,1,8] Aeotec 3-phase Meter / Phase 1 / Power
[3,50,2,0] Aeotec 3-phase Meter / Phase 2 / Energy
[3,50,2,8] Aeotec 3-phase Meter / Phase 2 / Power

9.3.4. New flows

As a result of the behaviour described in the *Low abstraction level* section (devices being asked to update their values) and in order to add actuation support, the flows have been reworked.

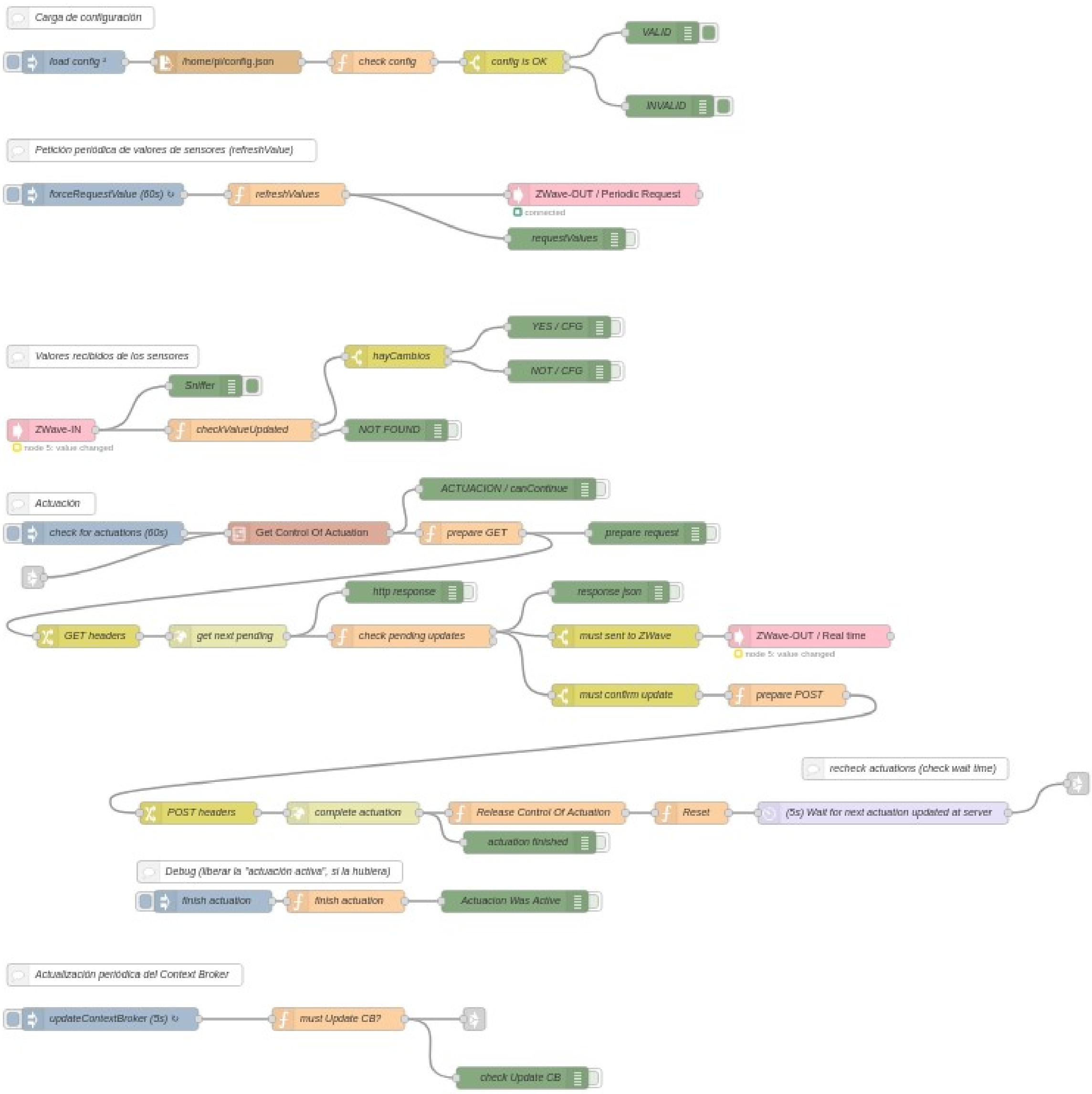


Figure 22. Z-Wave / Main flow

The main flow oversees:

- Loading the configuration.
- Refreshing the values of the devices (the *refreshValue* command is sent with the configured period for each attribute).
- Searching for pending actuations periodically.
- Executing the actuations if there's any pending (*setValue* command).
- Forwarding the received values with the configured period.

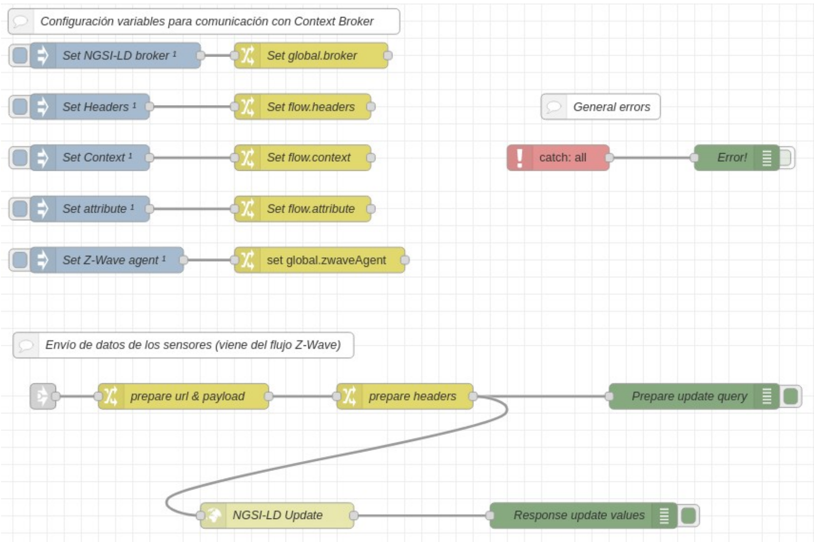


Figure 23. Z-Wave / Update the Context Broker

The previous flow is the one that actually sends the values to the Context Broker.

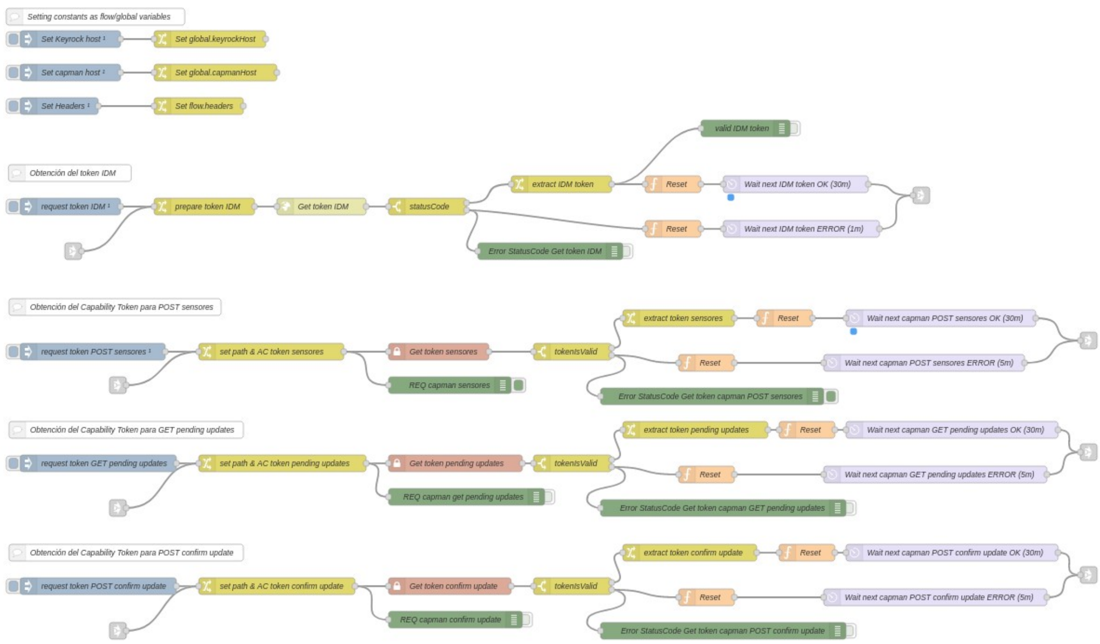


Figure 24. Z-Wave / Security

This flow is in charge of getting all the resources required to use the security components described in WP4, namely an identity token and authorization tokens for both sending information to the Context Broker and communicating with the Z-Wave actuation agent (described later in this document).

9.3.5. Configuration file

These flows use a json configuration file that maps Z-Wave attributes with entities. A sample file could be the next one:


```
{
	"urlpendingupdates":"https://phoenix.inf.um.es:1028/zwaveactuator/pendingUpdates",
	"urlconfirmupdate":"https://phoenix.inf.um.es:1028/zwaveactuator/confirmUpdate",
	"gw":"gw1234",
	"entities":[
		{"id":"1","nodeid":14,"cmdclass":37,"instance":1,"cmdidx":0,"isboolean":1}
	],
	"attributes":[
		{"id":"urn:ngsi-ld:Observation:UMU-Pleiades-BlockB-B1.1.014-ZWave5-Temperature","nodeid":5,"cmdclass":49,"instance":1,"cmdidx":1,"isboolean":0,"requiresconversion":0,"converttype":"","numdecimals":1,"pollingseconds":60,"nexttimestamppolling":0,"publishingseconds":300,"nexttimestamppublish":0,"units":"DegreeCelsius","minvalue":"-9","maxvalue":"50"},
		{"id":"urn:ngsi-ld:Observation:UMU-Pleiades-BlockB-B1.1.014-ZWave5-Humidity","nodeid":5,"cmdclass":49,"instance":1,"cmdidx":5,"isboolean":0,"requiresconversion":0,"converttype":"","numdecimals":0,"pollingseconds":60,"nexttimestamppolling":0,"publishingseconds":300,"nexttimestamppublish":0,"units":"Percent","minvalue":"0","maxvalue":"99"},
		{"id":"urn:ngsi-ld:Observation:UMU-Pleiades-BlockB-B1.1.014-ZWave5-CO2","nodeid":5,"cmdclass":49,"instance":1,"cmdidx":17,"isboolean":0,"requiresconversion":0,"converttype":"","numdecimals":0,"pollingseconds":60,"nexttimestamppolling":0,"publishingseconds":300,"nexttimestamppublish":0,"units":"PPM","minvalue":"0","maxvalue":"2000"}
	]
}
```


All the mappings between low level resources and entities are done here for both readings (*attributes*) and writings (*urlpendingupdates*, *urlconfirmupdate*, *gw* and *entities*). With this, the Context Broker doesn't need to know any information about the technology behind each device.

9.4. Intesis WMP

Intesis manufacturer sells universal IR transmitters/receivers (gateways) for splits compatible with different models from multiple manufacturers. Installed next to the splits, they receive the IR commands sent by the IR remote controller and this way every time the status of the split is changed, the gateway is updated too.

As external interface, for integration, the gateway has Wi-Fi support and an external software module can read and write attributes such as working mode, temperature set point, etc.

These devices have been integrated with an additional Node-Red flow of the Raspberry Pi image that shares the actuation and publication flows with the Z-Wave integration.

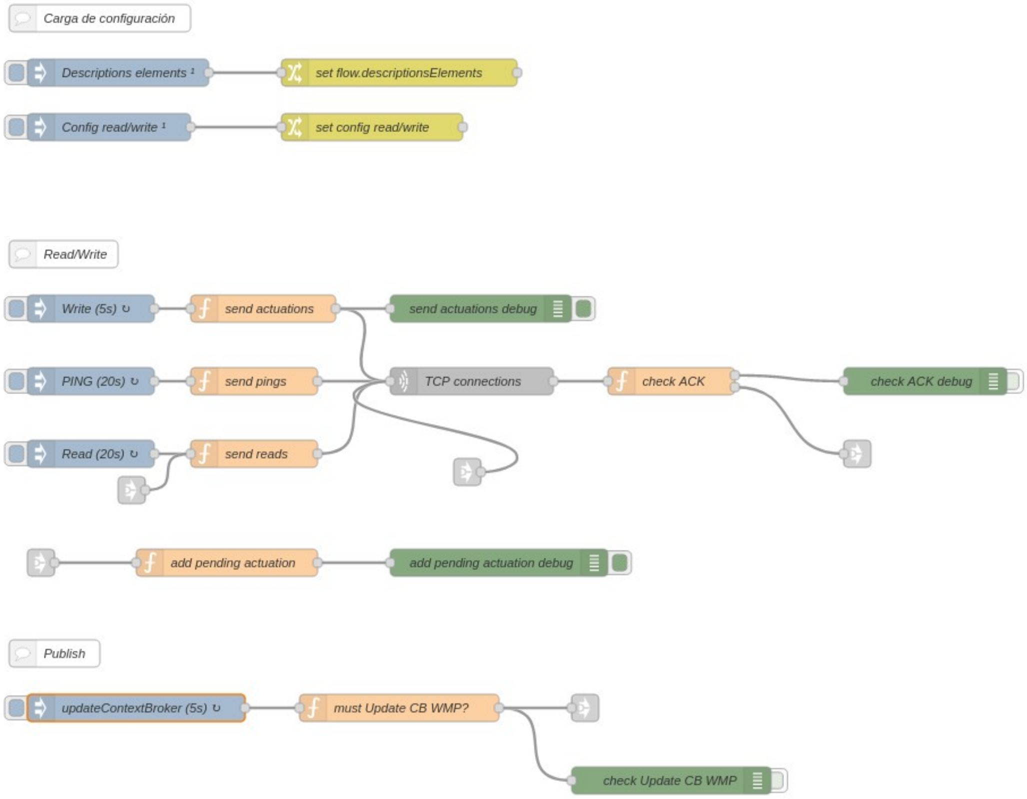


Figure 25. WMP / Main flow

For WMP support, an extra section has been added in the configuration file.

```
"wmp": [
  {
    "entities":[
      {"urn:ngsi-ld:Device:IR1-deviceState-Actuator":0},
      {"urn:ngsi-ld:Device:IR1-mode-Actuator":1},
      {"urn:ngsi-ld:Device:IR1-setPoint-Actuator":2},
      {"urn:ngsi-ld:Device:IR1-speed-Actuator":3}
    ],
    "attributes":[
      {"id":"urn:ngsi-ld:Observation:IR1-deviceState","units":"NoUnits","minvalue":"0","maxvalue":"1"},
      {"id":"urn:ngsi-ld:Observation:IR1-mode","units":"NoUnits","minvalue":"N/A","maxvalue":"N/A"},
      {"id":"urn:ngsi-ld:Observation:IR1-setPoint","units":"CelsiusDegrees","minvalue":"N/A","maxvalue":"N/A"},
      {"id":"urn:ngsi-ld:Observation:IR1-speed","units":"NoUnits","minvalue":"N/A","maxvalue":"N/A"}
    ],
    "ip":"192.168.1.100",
    "publishingseconds":300
  }
]
```

10. Data models & platform components

At the platform, multiple components have changed since D3.1, including different agents and other updates.

10.1. Connecting the Context Broker with the Historical data component

The connection between the Context Broker and the component that stores the historical data works in subscription mode, which means every update of an entity that's being registered by the Historical data component will trigger a notification from the Context Broker with the new value, or to be more accurate, new pair *value* + *timestamp*.

However a change was made after D3.1 once some random data loss was detected. By default devices try to align their readings, meaning a device reading every 5 minutes will send its values exactly on minutes 0, 5, 10 and so on. As a result most of the activity in the Context Broker will be concentrated in these specific moments with several devices sending data at the same time.

With the previous subscription strategy, the Historical data component only needed 1 subscription for all the entities and this communication pattern caused too much overhead for this single subscription and some of the notifications were lost.

The solution chosen to deal with this problem was to create one subscription per entity. It looks like the Context Broker can process multiple requests in parallel as long as they are linked to different subscriptions and the data loss problem is gone.

10.2. Tasmota agent

The integration of Smart Plugs that use the Tasmota firmware uses a custom agent that is running in the platform.

This agent adapts the Tasmota MQTT topic format to the standard MQTT topic format supported by the PHOENIX MQTT IoT-Agent for both reading values and executing commands. In addition the agent generates readings with the configured period exactly as any other device.

As indicated in D3.1, all the devices were configured so that the topic pattern in the Tasmota side looked like this:

*/PHOENIX-**PILOTNAME**/tasmotaPOWR2/**tasmotaX**/#*

So the agent is subscribed to one topic per pilot with the next subscription format (it will receive data from all the Tasmota devices of the pilot):

*/PHOENIX-**PILOTNAME**/tasmotaPOWR2/#*

The most important topics used by these devices are these:

/PHOENIX-**PILOTNAME**/tasmotaPOWR2/**tasmotaX**/STATE	Includes the POWER attribute (as well as other values)
/PHOENIX-**PILOTNAME**/tasmotaPOWR2/**tasmotaX**/SENSOR	Includes the power meter data
/PHOENIX-**PILOTNAME**/tasmotaPOWR2/**tasmotaX**/cmnd/POWER	ON/OFF commands are sent here
/PHOENIX-**PILOTNAME**/tasmotaPOWR2/**tasmotaX**/RESULT	Confirmations of the execution of commands are received here
/PHOENIX-**PILOTNAME**/tasmotaPOWR2/**tasmotaX**/POWER	The individual status of the POWER attribute is sent here

In the PHOENIX MQTT IoT-Agent side, each Tasmota device is converted to use the standard set of topics:

/PHOENIX-**PILOTNAME**/**tasmotaX**/attrs	Readings are received here
/PHOENIX-**PILOTNAME**/**tasmotaX**/cmd	Commands are sent here
/PHOENIX-**PILOTNAME**/**tasmotaX**/cmdexe	Confirmations of the execution of commands are received here

Here the agent is subscribed to one topic per pilot with the next subscription format (subscription to the commands sent to all the actuators of the pilot will be received and the agent will discard all those sent to non-Tasmota devices):
*/PHOENIX-**PILOTNAME**/+/cmd*

10.3. Actuation agent for Z-Wave and Intesis WMP devices

In order to provide actuation support for both Z-Wave and Intesis WMP devices, a common agent is deployed in the platform. The architecture for Z-Wave is shown in the next figure (Z-Wave interaction).

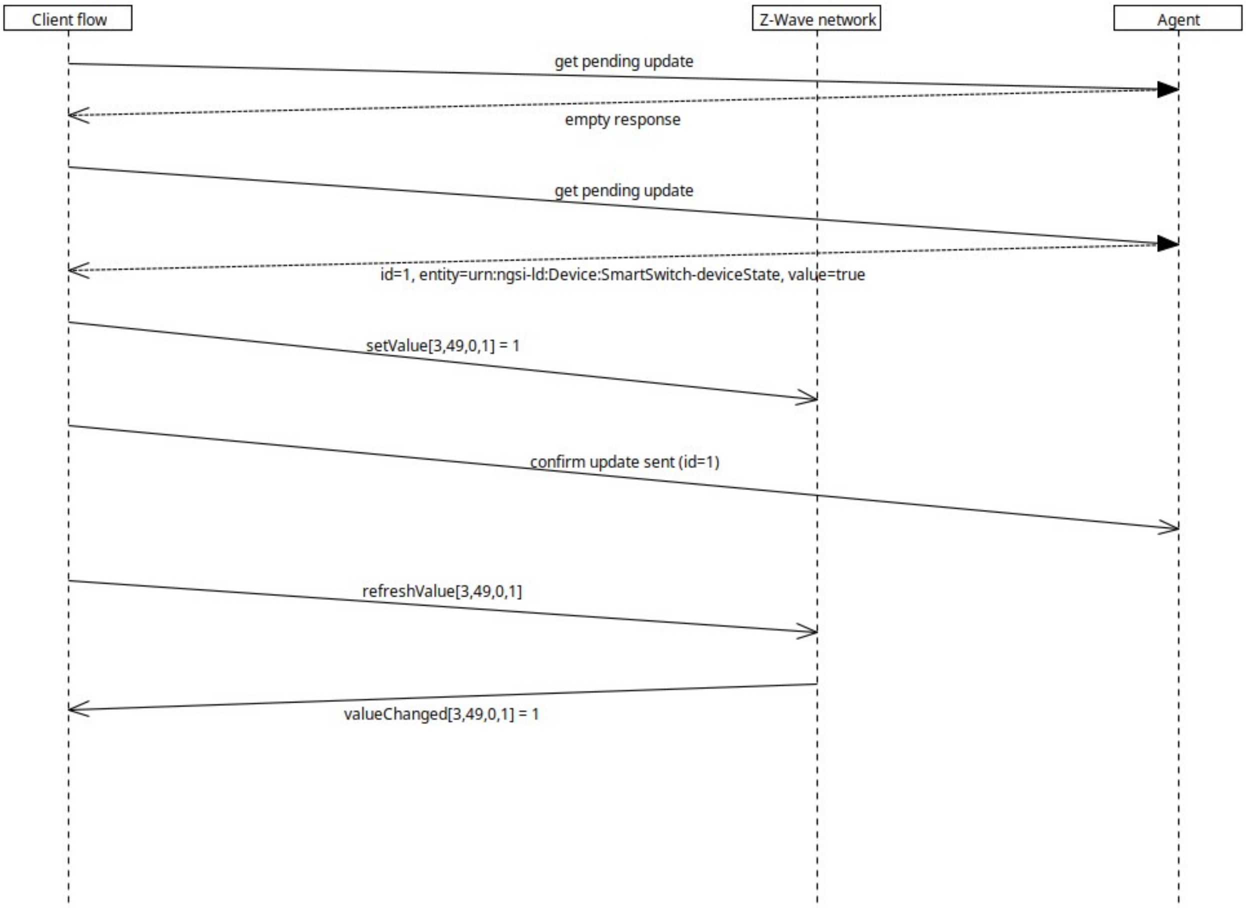


Figure 26. Actuation agent / Z-Wave

The client flow executed in the *Raspberry Pi* is checking for updates periodically and, when there's at least one pending, the flow will send the right command to the Z-Wave network based on its configuration and will tell the agent that the command has been sent. It doesn't mean the value will remain in the device until the next polling as it might be changed locally by a user after that (the *setValue* command gets no response). But if there's no such a local change, the new value will be received in the next polling sent by the main flow to the device (using the *refreshValue* command).

For WMP devices, the architecture is similar.

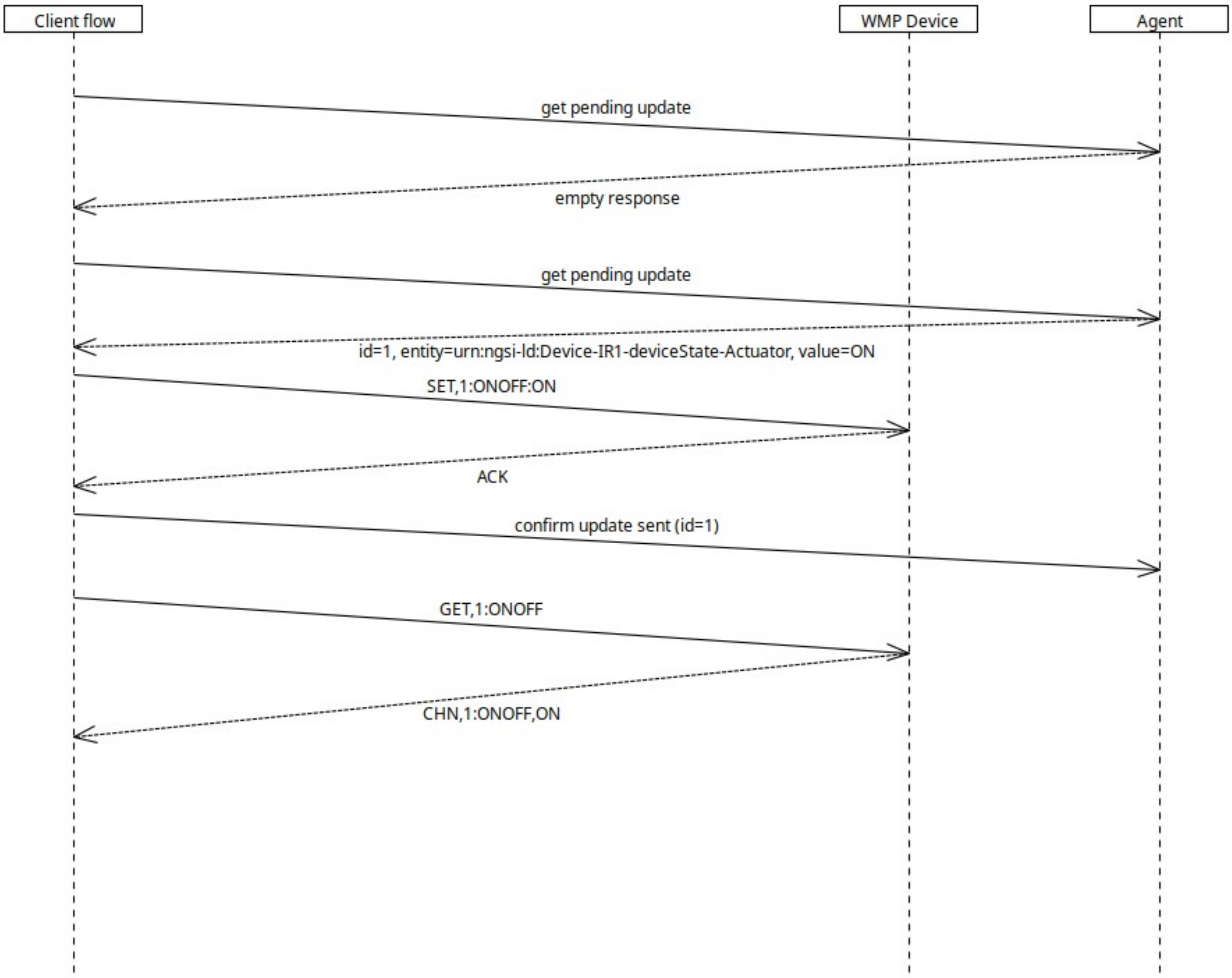


Figure 27. Actuation agent / WMP

For WMP devices the only difference is that the SET command is confirmed by the device (an ACK is received). The Agent ↔ Context Broker integration is based on the Context Provider concept.

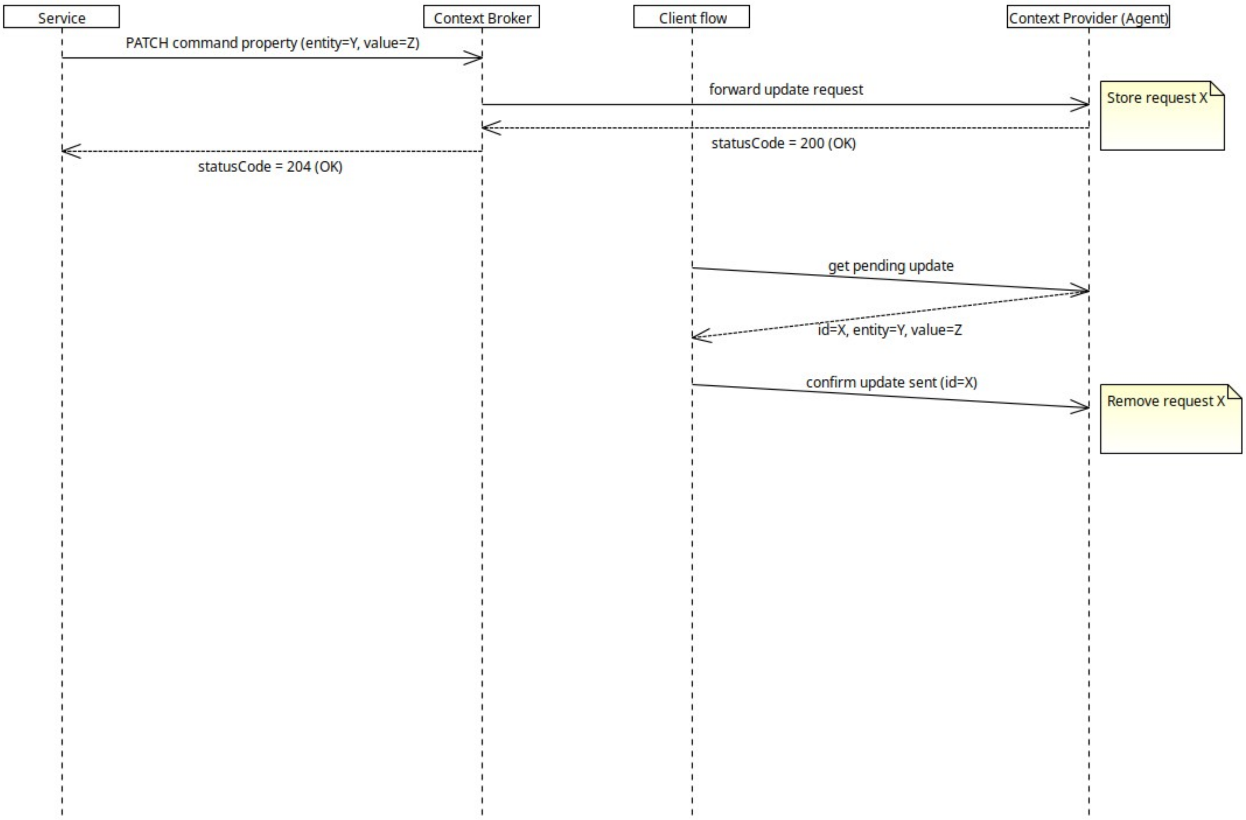


Figure 28. Actuation agent / Context Provider

The Context Broker can delegate the responsibility of generating portions of entities for both reading and writing to Context Providers. Here the Agent gets subscribed as a Context Provider in order to dynamically add the *command* property to all the actuation entities managed by it (Z-Wave and WMP actuation entities). When an update request (PATCH) is sent to any of any of these entities, the Agent will store the request and it will be sent to the involved gateway when it asks for it.

The way to register a new Context Provider is by sending a request like the next one.

```
POST http://phoenix.inf.um.es/ngsi-ld/v1/csourceRegistrations/urn:ngsi-ld:id-of-the-context-provider
Content-Type: application/ld+json
Accept: */*
fiware-service: phoenix
fiware-servicepath: /


{
  "@context": [
    "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
  ],
  "type": "ContextSourceRegistration",
  "name": "name of the provider",
```

```
  "description": "description of the provider",
  "information": [
    {
      "entities": [
        {
          "id":"urn:ngsi-ld:Device:device1",
          "type":"https://uri.fiware.org/ns/data-models#Device"
        },
        {
          "id":"urn:ngsi-ld:Device:device2",
          "type":"https://uri.fiware.org/ns/data-models#Device"
        }
      ]
      "properties": [
        "command"
      ]
    }
  ],
  "endpoint": "http://your.ip.address:port"
}
```


10.4. Format of new types of entities

In addition to the previous *Building* → *Zone* → *Device* → *IotStream* → Observation hierarchy described in D3.1, new entities have been defined to cover the requirements of the new content integrated with the platform.

10.4.1. Actuators

Actuators are modelled as special Devices that provide one or more *command* properties which modify certain aspects that are controlled/observed by *normal* Devices. If no other change is made after an actuation is executed, the value sent should be available in the right Observation once the next reading from the device is received. So those *normal* Devices represent the read-only half of the Device (the category is usually *sensor*) while actuators represent the write-only half.

An actuation entity could have the following format.


```
{
  "@context": "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
  "id": "urn:ngsi-ld:Device:UMU-Pleiades-BlockB-B1.1.017-SmartPlug2-CommandOutputOnOff",
  "type": "https://uri.fiware.org/ns/data-models#Device",
  "valve1on_status": {
    "value": {
      "@type": "commandStatus",
      "@value": "OK"
    },
```

```
    "type": "Property",
    "observedAt": "2021-10-19T13:26:57.061Z"
  },
  "valve1on_info": {
    "value": {
      "@type": "commandResult",
      "@value": "Complete"
    },
    "type": "Property",
    "observedAt": "2021-10-19T13:26:57.061Z"
  },
  "valve1off_status": {
    "value": {
      "@type": "commandStatus",
      "@value": "OK"
    },
    "type": "Property",
    "observedAt": "2021-10-19T13:27:15.255Z"
  },
  "valve1off_info": {
    "value": {
      "@type": "commandResult",
      "@value": "Complete"
    },
    "type": "Property",
    "observedAt": "2021-10-19T13:27:15.255Z"
  },
  "description": {
    "value": "Command ON/OFF SmartPlug 2 Pleiades Block B / B1.1.017",
    "type": "Property"
  },
  "https://uri.fiware.org/ns/data-models#category": {
    "value": [
      "actuator"
    ],
    "type": "Property"
  },
  "https://uri.fiware.org/ns/data-models#controlledProperty": {
    "value": [
      "valve1on",
      "valve1on_info",
      "valve1on_status",
      "valve1off",
      "valve1off_info",
      "valve1off_status"
    ],
```

```
      "type": "Property"
   },
   "https://uri.fiware.org/ns/data-models#Commands": {
      "value": [
         {
            "name": "valve1on",
            "description": "Switch on the smart plug",
            "observedProperty": "urn:ngsi-ld:controlledProperty:deviceState",
            "refDevice": "urn:ngsi-ld:Device:UMU-Pleiades-BlockB-B1.1.017-SmartPlug2-StatusOutputOnOff"
         },
         {
            "name": "valve1off",
            "description": "Switch off the smart plug",
            "observedProperty": "urn:ngsi-ld:controlledProperty:deviceState",
            "refDevice": "urn:ngsi-ld:Device:UMU-Pleiades-BlockB-B1.1.017-SmartPlug2-StatusOutputOnOff"
         }
      ],
      "type": "Property"
   },
   "valve1on": {
      "type": "command",
      "value": ""
   },
   "valve1off": {
      "type": "command",
      "value": ""
   }
}
```


The key points here are:

- For each command, there is a property of type *command* with the name of the command itself (**valve1on** and **valve1off** in this example).
- For each command, two additional properties are created with the info and the status of the execution (**valve1on_info**, **valve1on_status**, **valve1off_info** and **valve1off_status**).
- For each command, one entry is included in the array of **Commands** indicating the name of the command, the reference to the *normal* Device indirectly updated by the command and the property involved (one of the *controlledProperties* of the Device, which is used to get access to the right Observation through its *observedProperty*).

With this definition one actuation entity can control multiple Devices and this is something important when these Devices are using the PHOENIX MQTT IoT-Agent because for this agent all the commands that must be sent to the same *DEVICE_ID* (according to the MQTT topic format) must be in the same entity. This is especially useful for BMS integrations where one *DEVICE_ID* (and therefore only one subscription from the BMS side) is enough to receive the commands from all the devices the BMS is controlling.

10.4.2. Entsoe

For the Entsoe integration, two custom types of entities have been defined. The first one is used for storing energy-related values from different sets of readings.


```
{
  "@context": "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
  "id": "urn:ngsi-ld:DailyPerHourEnergyReading:Spain-TotalLoad",
  "type": "http://phoenix.inf.um.es/data-models/DailyPerHourEnergyReading-schema.jsonld",
  "areaServed": {
    "type": "Property",
    "value": "Spain, REE BZ / CA / MBA"
  },
  "https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/category": {
    "type": "Property",
    "value": "Load"
  },
  "source": {
    "type": "Property",
    "value": "All"
  },
  "http://www.w3.org/ns/sosa/hasResult": {
    "value":
"[24726,23199,22335,22076,22187,22966,26280,30783,33145,33716,34002,33786,33605,33340,32742,32237,32151,32700,34242,34982,
35360,34448,31571,28521]",
    "type": "Property",
    "observedAt": "2021-12-12T23:00:00.000Z",
    "http://qudt.org/1.1/schema/qudt#unit": {
      "type": "Property",
      "value": "http://qudt.org/1.1/vocab/unit#Megawatthour"
    }
  }
}
```


In every update of the entity, an array of readings in String format will be set as value with one reading per hour. The timestamp of the combined reading is set at the beginning of the day.

The second one is used for storing prices in the set of day-ahead prices.

```
{
  "@context": "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
  "id": "urn:ngsi-ld:DailyPerHourPrices:Spain-DayAheadPrices",
  "type": "http://phoenix.inf.um.es/data-models/DailyPerHourPrices-schema.jsonld",
  "description": {
    "type": "Property",
    "value": "Day-ahead Prices for Spain's market"
  },
  "http://www.w3.org/ns/sosa/hasResult": {
    "value":
"[290.98,283.89,276.8,254.24,250,265.45,290.16,294.32,298.04,295.86,289.85,270.65,280,284.13,281.23,289.74,292.78,301.07,315.13,31
9.13,320,299.41,292.78,270.99]",
    "type": "Property",
    "observedAt": "2021-12-13T23:00:00.000Z",
    "https://schema.org/priceCurrency": {
      "type": "Property",
      "value": "EUR"
    }
  }
}
```


10.4.3. REE and Irish pricing market

For the integrations of the Spanish and the Irish energy pricing markets, a custom type of entity has been defined.


```
{
  "@context": "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
  "id": "urn:ngsi-ld:EnergyOffer:1MegaWattHour_ree",
  "type": "https://schema.org/Offer",
  "https://schema.org/priceCurrency": {
    "type": "Property",
    "value": "EUR"
  },
  "https://schema.org/includesObject": {
    "type": "Property",
    "value": "",
    "https://schema.org/amountOfThisGood": {
      "type": "Property",
      "value": "1"
    },
    "http://qudt.org/1.1/schema/qudt#unit": {
      "type": "Property",
      "value": "http://qudt.org/1.1/vocab/unit#Megawatthour"
    }
```

```
  },
  "https://schema.org/price": {
    "value": 418.09,
    "type": "Property",
    "observedAt": "2021-12-14T17:00:00.000Z"
  }
}
```

Every hour the entities are updated with the official price of the energy in this hour.

10.4.4. EV chargers

For the integration of the EV chargers, an extended version of the Observation entity has been defined.

```
{
  …
  "http://www.w3.org/ns/sosa/hasResult": {
    "type": "Property",
    "value": "1234.567",
    "observedAt": "2021-12-16T15:43:46.636Z",
    "dateObservedFrom": {
      "type": "Property",
      "value": "2021-12-14T17:00:51Z"
    },
    "dateObservedTo": {
      "type": "Property",
      "value": "2021-12-14T17:15:51Z"
    }
  }
}
```

The UMU Scada, in addition to the amount of energy consumed by the vehicle, also provides both the beginning and the end timestamps of the charge cycle, which are stored in new properties.

10.5. Templates

In the context of provisioning entities based on the already-described hierarchy, a base template (in spreadsheet format) has been filled by pilot partners with the existing Buildings, Devices, etc. in their pilot.

Each template also includes extra columns that are used for generating configuration and deployment files for both gateways and agents. Each template includes the next sections (compact format).

Name	Type	Description	Must insert?	Address.Locality	Address.PostalCode	Address.StreetAddress
KAMA-Building	resident	Block of flats	0	Thessaloniki	55133	Dalipi Camp

Figure 29. Templates / Buildings

Name	Type	Building	Floor	Must insert?
KAMA-Building-Flat1	apartment	KAMA-Building	0	0
KAMA-Building-Flat2	apartment	KAMA-Building	0	0
KAMA-Building-Flat3	apartment	KAMA-Building	0	0
KAMA-Building-Flat4	apartment	KAMA-Building	0	0
KAMA-Building-Flat5	apartment	KAMA-Building	1	0
KAMA-Building-Flat6	apartment	KAMA-Building	1	0
KAMA-Building-Flat7	apartment	KAMA-Building	1	0
KAMA-Building-Flat8	apartment	KAMA-Building	1	0

Figure 30. Templates / Zones

Name	Description	Zone
KAMA-Building-Common-Multisensor	Multisensor 6 for ambient light, humidity, temperature	KAMA-Building-Common
KAMA-Building-Common-CO2Monitor	MCO Home MH9-CO2 for building's CO2 concentration, humidity, temperature (internal)	KAMA-Building-Common
KAMA-Building-Common-FroniusSymo20	Inverter for the solar production power and energy	KAMA-Building-Common
KAMA-Building-Common-FroniusGEN24	Hybrd inverter for the state of the battery	KAMA-Building-Common
KAMA-Building-Common-SmartImplant1.1	Temperature sensors for solar thermal boiler temperature IN of apartment 1	KAMA-Building-Common
KAMA-Building-Common-SmartImplant1.2	Temperature sensors for solar thermal boiler temperature OUT of apartment 1	KAMA-Building-Common
KAMA-Building-Common-SmartImplant1.3	Temperature sensors for solar thermal boiler temperature IN of apartment 2	KAMA-Building-Common
KAMA-Building-Common-SmartImplant1.4	Temperature sensors for solar thermal boiler temperature OUT of apartment 2	KAMA-Building-Common
KAMA-Building-Common-SmartImplant1.5	Temperature sensors for solar thermal boiler temperature IN of apartment 3	KAMA-Building-Common
KAMA-Building-Common-SmartImplant1.6	Temperature sensors for solar thermal boiler temperature OUT of apartment 3	KAMA-Building-Common

Figure 31. Templates / Devices

Name	Property name	Units	MinValue	MaxValue	Reading Period
KAMA-Building-Common-Multisensor	light	http://qudt.org/vocab/unit#Lux	0	30000	5 minutes
KAMA-Building-Common-Multisensor	humidity	http://qudt.org/1.1/vocab/unit#Percent	20	90	5 minutes
KAMA-Building-Common-Multisensor	temperature	http://qudt.org/1.1/vocab/unit#DegreeCelsius	-10	50	5 minutes
KAMA-Building-Common-CO2Monitor	co2	http://qudt.org/1.1/vocab/unit#PPM	0	2000	5 minutes
KAMA-Building-Common-CO2Monitor	humidity	http://qudt.org/1.1/vocab/unit#Percent	0	99	5 minutes
KAMA-Building-Common-CO2Monitor	temperature	http://qudt.org/1.1/vocab/unit#DegreeCelsius	-9	50	5 minutes
KAMA-Building-Common-FroniusSymo20	totalActiveEnergyExport	http://qudt.org/1.1/vocab/unit#Kilowatthour	0	Infinite	15 minutes
KAMA-Building-Common-FroniusSymo20	frequency	http://qudt.org/1.1/vocab/unit#Hertz	45	65	5 minutes
KAMA-Building-Common-FroniusGEN24	totalActiveEnergyExport	http://qudt.org/1.1/vocab/unit#Kilowatthour	0	Infinite	15 minutes
KAMA-Building-Common-FroniusGEN24	frequency	http://qudt.org/1.1/vocab/unit#Hertz	45	66	5 minutes
KAMA-Building-Common-FroniusGEN24	batteryState	http://qudt.org/1.1/vocab/unit#Percent	0	100	5 minutes
KAMA-Building-Common-FroniusGEN24	setPoint	http://qudt.org/1.1/vocab/unit#Percent	5	100	5 minutes

Figure 32. Templates / Properties of Devices (information)

Name	Property name	Is Z-Wave?	nodeid	cmdclass	instance	cmdidx	isboolean	Use multiplier	multiplier	Requires conversion	Conversion	Units Compact
KAMA-Building-Common-Multisensor	light	1	4	49	1	3	0	0		0		Lux
KAMA-Building-Common-Multisensor	humidity	1	4	49	1	5	0	0		0		Percent
KAMA-Building-Common-Multisensor	temperature	1	4	49	1	1	0	0		1	toCelsius	DegreeCelsius
KAMA-Building-Common-CO2Monitor	co2	1	6	49	1	17	0	0		0		PPM
KAMA-Building-Common-CO2Monitor	humidity	1	6	49	1	5	0	0		0		Percent
KAMA-Building-Common-CO2Monitor	temperature	1	6	49	1	1	0	0		1	toCelsius	DegreeCelsius

Figure 33. Templates / Properties of Devices (configuration)

Name	Master Device Name	Command name	Observed property
KAMA-Building-Flat1-SmartPlug1-Actuator	KAMA-Building-Flat1-SmartPlug1-InputStatus	command	deviceState
KAMA-Building-Flat2-SmartPlug2-Actuator	KAMA-Building-Flat2-SmartPlug2-InputStatus	command	deviceState
KAMA-Building-Flat3-SmartPlug3-Actuator	KAMA-Building-Flat3-SmartPlug3-InputStatus	command	deviceState
KAMA-Building-Flat4-SmartPlug4-Actuator	KAMA-Building-Flat4-SmartPlug4-InputStatus	command	deviceState
KAMA-Building-Flat5-SmartPlug5-Actuator	KAMA-Building-Flat5-SmartPlug5-InputStatus	command	deviceState
KAMA-Building-Flat6-SmartPlug6-Actuator	KAMA-Building-Flat6-SmartPlug6-InputStatus	command	deviceState
KAMA-Building-Flat7-SmartPlug7-Actuator	KAMA-Building-Flat7-SmartPlug7-InputStatus	command	deviceState
KAMA-Building-Flat8-SmartPlug8-Actuator	KAMA-Building-Flat8-SmartPlug8-InputStatus	command	deviceState

Figure 34. Templates / Actuators

Name	Context	Comments
urn:ngsi-ld:EnergyOffer:1MegaWattHour_ree	REE integration	1 reading per hour (pricing in €)
urn:ngsi-ld:DailyPerHourEnergyReading:Spain-TotalLoad	Entsoe integration (MWH) N Generation types and 4 countries: - Spain. Area code **Spain, REE BZ / CA / MBA** - Greece. Area code **Greece, IPTO BZ / CA/ MBA** - Ireland. Area code **Ireland (SEM) BZ / MBA** - Sweden. Area code **SE2 BZ / MBA**	Each day, from 23:00 to 23:10 local time in each country, the agent reads:- TotalLoad from the previous day- TotalLoadForecast (DayAhead) fo
urn:ngsi-ld:DailyPerHourEnergyReading:Spain-TotalLoadDayAhead		
urn:ngsi-ld:DailyPerHourEnergyReading:Spain-GenerationForecast-Wind		
urn:ngsi-ld:DailyPerHourEnergyReading:Spain-GenerationForecast-Solar		
urn:ngsi-ld:DailyPerHourEnergyReading:Spain-Generation-XXX		
urn:ngsi-ld:DailyPerHourPrices:Spain-DayAheadPrices		

Figure 35. Templates / Other Devices

Using a set of scripts on each template, what is generated is:

- The set of base entities in cascade (Buildings, Zones inside Buildings, Devices inside Zones, etc.).
- Actuation entities (with the references they have to other Devices and indirectly to Observations).
- Other entities (Entsoe, REE, etc.).
- Subscriptions between the Context Broker and the Historical data component.
- Subscriptions for the PHOENIX MQTT IoT-Agent.
- Configuration files for the service executed at the gateways that controls Modbus devices.
- Configuration files for Node-Red for Z-Wave and WMP devices.

11. Framework for SRI Performance Indicators

This section describes framework of the project for the Smart Readiness Indicator (SRI) Performance indicators. Within the project, the SRI scores have been considered the best way to evaluate the effectiveness of the ICT solutions proposed in the project, in terms of smartness of legacy systems and appliances in the pilot buildings. This has been translated into two main Tasks of the project: Task 3.4 is dedicated to the creation of a framework for SRI performance indicator, and Task 5.3 aims to make automatic the SRI calculation.

Since the beginning of the project, the PHOENIX consortium has worked towards having a common protocol for the SRI evaluation. A webinar was organised for this reason in January 2021, to explain the use and implementation of the SRI within project PHOENIX. Through the webinar, all the partners received theoretical and practical information about the SRI methodology and the calculation tool. Also, the opportunities for PHOENIX have been extensively discussed. Therefore, in February 2021 all the partners were asked to deliver an SRI calculation for the actual state of the pilots, i.e. considering the smartness of the building before the intervention. In this first phase, to have an easier approach to the tool and also to have the opportunity to test both methods over the duration of the project, the partners have used the simplified spreadsheet, the works with the so-called Method A. Results are shown in Table 13.

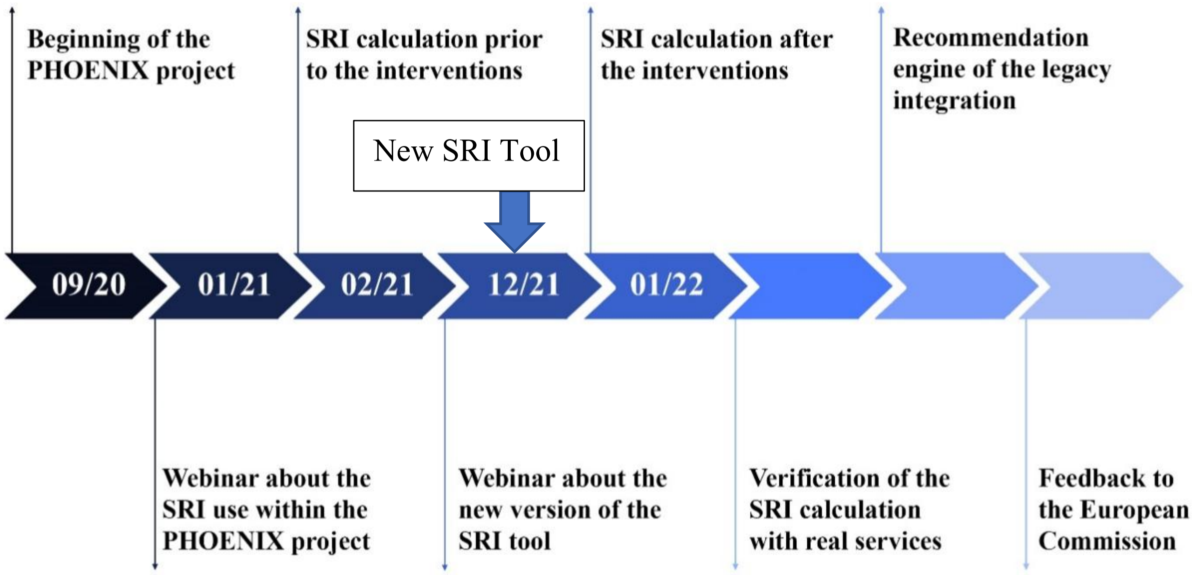


Figure 36 Timeline of the SRI framework in the PHOENIX project

Table 13 SRI scores of the pilots building within the project PHOENIX

Pilot	Building Type	Climate Zone	SRI score
Arden #1	Residential	West Europe	5%
Arden #2	Residential	West Europe	5%
Arden #3	Non-Residential	West-Europe	27%
KAMA	Residential	South Europe	17%
LTU	Residential	North Europe	13%
MIW #1	Non-Residential	South Europe	8%
MIW #2	Residential	South Europe	9%
UMU	Non-residential	South Europe	12%

The interest of the consortium in the SRI tool has been continuous since then, leading some of the partners to prepare an estimation of the expected SRI score after the intervention. Once the partners got a full comprehension of the tool, the SRI score started to be calculated with Method B, more detailed and more appropriate for complex buildings like the ones used as pilots.

At the end of 2021, a new version of the tool was launched, and the consortium of the project PHOENIX offered to test it. A new webinar was necessary to discuss the potentialities of the new version. In particular, catalogues of services of Method A and Method B can be modified to adapt to the real conditions of each pilot, since some of the services will be not present or not applicable to a particular building. Furthermore, in the new version, it is possible to work with a customised catalogue of services (the so-called Method Custom Service Mix), that allows adding extra services to the catalogue. Therefore, in January 2022 the partners are asked to 1) prepare an SRI calculation with the full catalogue of Method B that refers to the situation after the intervention and 2) prepare a customised SRI calculation, choosing from the catalogue of Method B only the services that take into account of the specificities of each building (Figure 37).

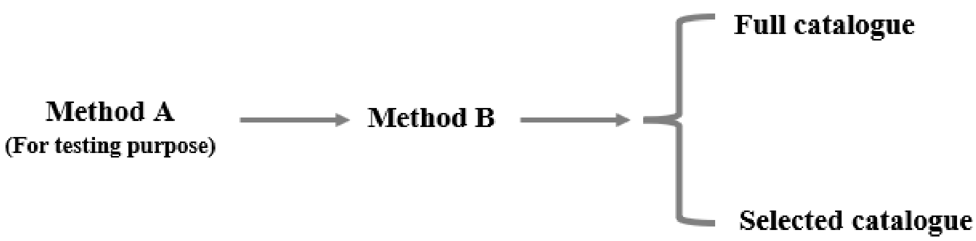


Figure 37. Approach to the SRI methodology within the project PHOENIX

The framework includes the next steps that the consortium will face over 2022. Firstly, once the intervention on the pilots building have been finalised, the partner will have to verify that the SRI obtained with the real services correspond to the expected SRI previously calculated. The phase of verification will be made partly automatic since the PHOENIX Smartness Hub will be connected with the smart devices of the controllable services. Within the project there will be also developed a recommendation engine of the legacy integration. During the project, all the results and feedback from the testing will be reported to EASME and the SRI developers.

12. References

[1] www.rediscoverycentre.ie

[2] Entsoe Transparency portal https://transparency.entsoe.eu/

[3] REE https://www.ree.es/es/apidatos

[4] D-Bus https://www.freedesktop.org/wiki/Software/dbus/

[5] CAN BUS https://en.wikipedia.org/wiki/CAN_bus

[6] CANopen https://en.wikipedia.org/wiki/CANopen

Annex I – Enteliweb API for BMS Connectivity

The Enteliweb software provides a wide array of API functions, the comprehensive documentation of these function is listed in the resources section of this folder. This manual provides a detailed description of how to extract data from sensors set up using the Enteliweb Building Management Software (BMS).

The Enteliweb BMS has been set up in the Rediscovery Centre (RDC) "a state of the art facility with a focus on sustainability and reuse" [1]. The facility has a CHP, boiler, solar panels and a heat pump which are used to generate heat and electricity for the buildings. It also has several temperature sensors placed in strategic locations around the building.

GENERAL API REQUEST STRUCTURE

The base structure of an Enteliweb API request is as follows,

Format	:	http://<server IP address>/enteliweb/api
Example	:	http://localhost/enteliweb/api

Local host should be replaced with the server IP address if Enteliweb is being accessed using a remote server (local host is used when accessing Enteliweb on the same computer it has been installed on).

It is important to note that the API request made will need a username and password associated with it (this can be manually entered when sending the API request) or can be preconfigured in a coding script. The username and password are consistent with the Enteliweb log in details.

In order to obtain the list of sites set up using Enteliweb the following request should be made.

Format	:	http://<server IP address>/enteliweb/api/.bacnet
Example	:	http://localhost/enteliweb/api/.bacnet
Response	:	


```
<Collection xmlns="http://bacnet.org/csml/1.2" nodeType="PROTOCOL">
       <Collection name="Ballymun_BH" nodeType="NETWORK" truncated="true"/>
</Collection>
```

There exists only one site in the RDC, Ballymun_BH, this code name will be required to make subsequent API requests from this site.

In order to obtain a list of all the devices registered on a site the following API request must be made.

Format	:	http://<server IP address>/enteliweb/api/.bacnet/<site name>
Example	:	http://localhost/enteliweb/api/.bacnet/Ballymun_BH
Response	:	

```
<Collection xmlns="http://bacnet.org/csml/1.2">
<Collection name="50016" displayName="Photovoltaic" nodeType="DEVICE" truncated="true"/>
<Collection name="50017" displayName="Lighting" nodeType="DEVICE" truncated="true"/>
<Collection name="2000" displayName="BH_Controller" nodeType="DEVICE" truncated="true"/>
</Collection>
```

DATA QUERIES

Time based data is stored in the Enteliweb software as “Trend Logs” these trend logs can be accessed via a simple API request. To find a complete list of all available trend logs on the site the following API request must be made.

Format	:	http:// <server IP address>//enteliweb/api/.data/histories
Example	:	http://localhost/enteliweb/api/.data/histories
Response	:	

```
<List xmlns="http://bacnet.org/csml/1.2" name="histories">
<Link name="1" value="/.bacnet/Ballymun_BH/2000/trend-log,77"/>
<Link name="2" value="/.bacnet/Ballymun_BH/2000/trend-log,78"/>
<Link name="3" value="/.bacnet/Ballymun_BH/2000/trend-log,79"/>
</List>
```

The names of the sensors that the trend logs represent can be found as follows. Their names can be determined by taking the device ID from the path containing the trend logs. For example, the

"2000" device ID (referenced above) contains all the trend logs for the RDC sight. The names of these trend logs can therefore be accessed via the following API request.

Format : http:// <server IP address>//enteliweb/api/.bacnet/<site name >/<device name>

Example : http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000

Response :


```
<Collection xmlns="http://bacnet.org/csml/1.2" nodeType="Device">
<Object name="trend-log,1" displayName="CHP_Flow Temp TL1" truncated="true"/>
<Object name="trend-log,2" displayName="CHP_Retn Temp TL2" truncated="true"/>
<Object name="trend-log,3" displayName="Boiler_Flow Temp TL3" truncated="true"/>
<Object name="trend-log,4" displayName="Boiler_Retn Temp TL4" truncated="true"/>
</Collection>
```


A table can then be created to show each trend log number and its corresponding name and API request URL address.

To access data entries from the trend logs a time range should be specified. The RDC trend logs give 1 data measurement every 5 minutes and approximately 3 days' worth of data entries can be accessed from 1 API request.

The time range is specified in the API request as follows,

Format :

http://localhost/enteliweb/api/.bacnet/<site name> /<device name>/<trend log number> /
log-buffer?published-ge=<start date>&published-le=<end date>

Example :

http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,135/
log-buffer?published-ge=2021-11-18T00:00:00&published-le=2021-11-18T13:15:00

Response :


```
<List xmlns="http://bacnet.org/csml/1.2" name="log-buffer">
<Sequence name="1">
<DateTime name="timestamp" value="2021-11-18T13:00:00.70"/>
<Choice name="logDatum">
<Real name="real-value" value="4.89377117"/>
```

```
</Choice>
<BitString name="statusFlags" value="0"/>
</Sequence>
<Sequence name="2">
<DateTime name="timestamp" value="2021-11-18T13:05:00.60"/>
<Choice name="logDatum">
<Real name="real-value" value="4.96163321"/>
</Choice>
<BitString name="statusFlags" value="0"/>
</Sequence>
<Sequence name="3">
<DateTime name="timestamp" value="2021-11-18T13:10:00.40"/>
<Choice name="logDatum">
<Real name="real-value" value="4.95440912"/>
</Choice>
<BitString name="statusFlags" value="0"/>
</Sequence>
<Sequence name="4">
<DateTime name="timestamp" value="2021-11-18T13:15:00.80"/>
<Choice name="logDatum">
<Real name="real-value" value="4.95833397"/>
</Choice>
<BitString name="statusFlags" value="0"/>
</Sequence>
</List>
```


The time range was set to obtain data from 1:00pm to 1:15pm on Friday November 18th 2021. The trend log number was set to 135 which corresponds to the lighting power consumption in kW. This value was measured and recorded every 5 minutes. The timestamp value is found next to the “timestamp” string indicator and the lighting power consumption value is found next to the “real-value” string indicator.

A simple algorithm can be written to automatically save these values and generate a plot of the data values and timestamps on a given day. This process has been completed on a prespecified day for every trend log contained within the RDC Enteliweb database (133 graphs in total)

The script to extract these values and generate the plots is found in the repository section of this folder. The corresponding generated plots can be found in the “plots” folder. Several of the more interesting plots have been included in the graphs section of this document.

WRITING DATA

The process required for setting data is slightly more complex, firstly the data type of the object you are setting must be determined. This can be found by sending a get request for the property of the device to which you are looking to set a value.

Format :

```
GET http://<server IP address>/enteliweb/api/.bacnet/<site name>/<device number>/< object type>,<instance>/<property name>/<sub-property path>
```

Example :

```
GET http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/binary-value,5/present-value
```

Response :

```
<Enumerated xmlns="http://bacnet.org/csml/1.2" value="inactive"/>
```

Therefore, in order to set the value type for this object the format of the body of the message is as follows

```
Body : <Enumerated xmlns="http://bacnet.org/csml/1.2" value=<desired value>/>
```

The format and example request and response of a put request of this type is outlined below.

Format :

```
URL : PUT http://<server IP address>/enteliweb/api/.bacnet/<site name>/<device number>/< object type>,<instance>/<property name>/<sub-property path>

Body : <<datatype> xmlns="http://bacnet.org/csml/1.2" value="<value>"/>
```

Example :

```
URL  :  PUT http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/binary-value,5/present-value
Body : <Enumerated xmlns="http://bacnet.org/csml/1.2" value="inactive"/>
```

Response :

```
<Enumerated error="-1" errorText="OK" xmlns="http://bacnet.org/csml/1.2"/>
```

The “OK” value indicates the value was successfully set.

Keywords

Binary value types have two possible values

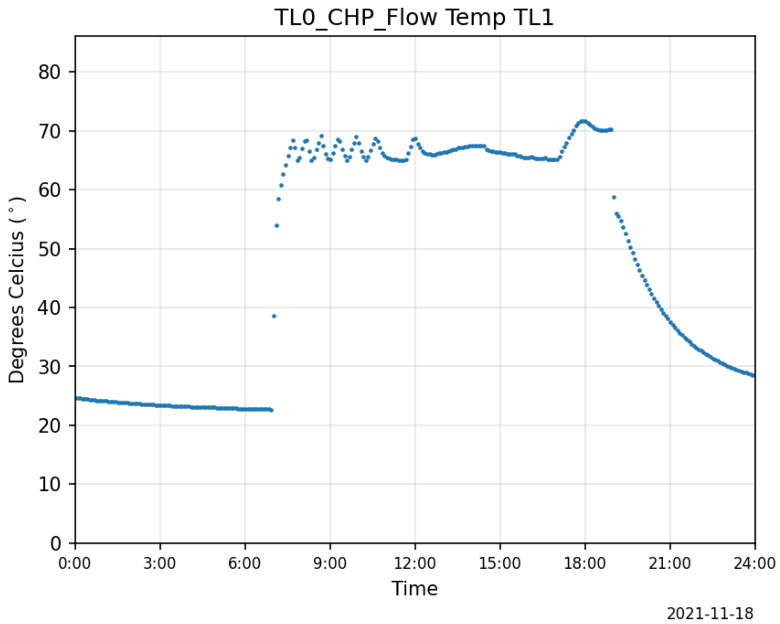
- active
- inactive

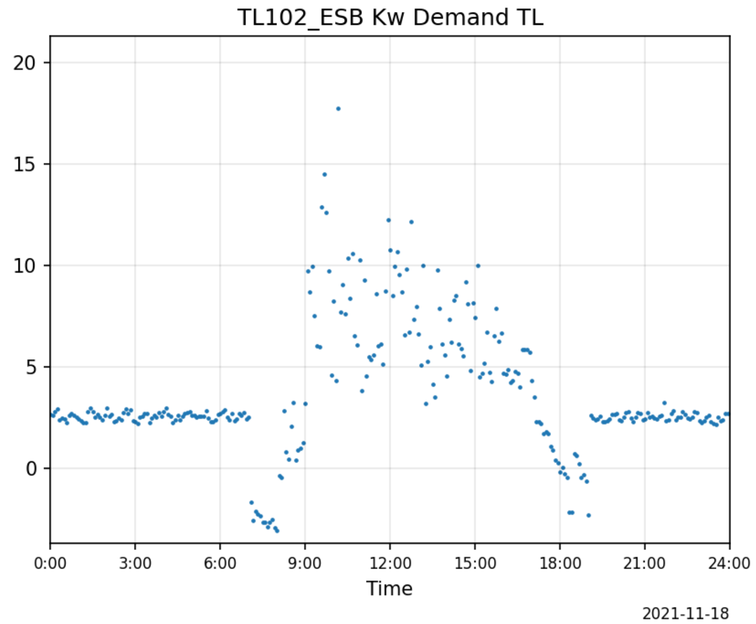
Analog value types just take a numeric value as input. In order to determine the units used by a given analog property add "/units" to the end of the property request made property. An example request is outline below.

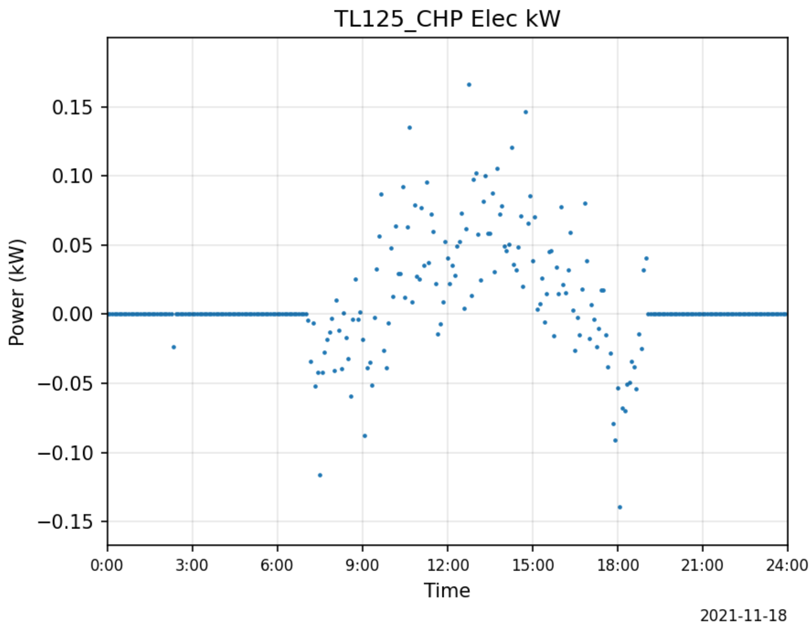
http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/analog-input,1501/units

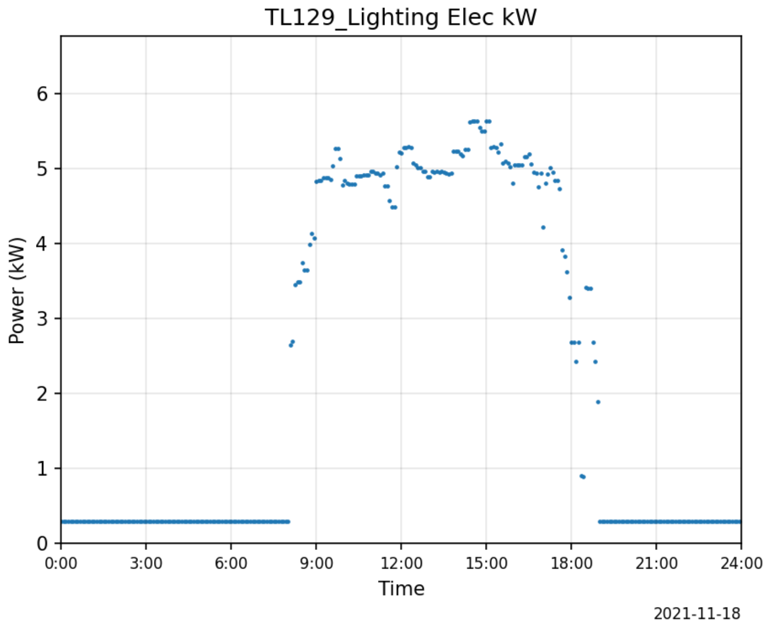
REDISCOVERY CENTRE DATA PLOTS

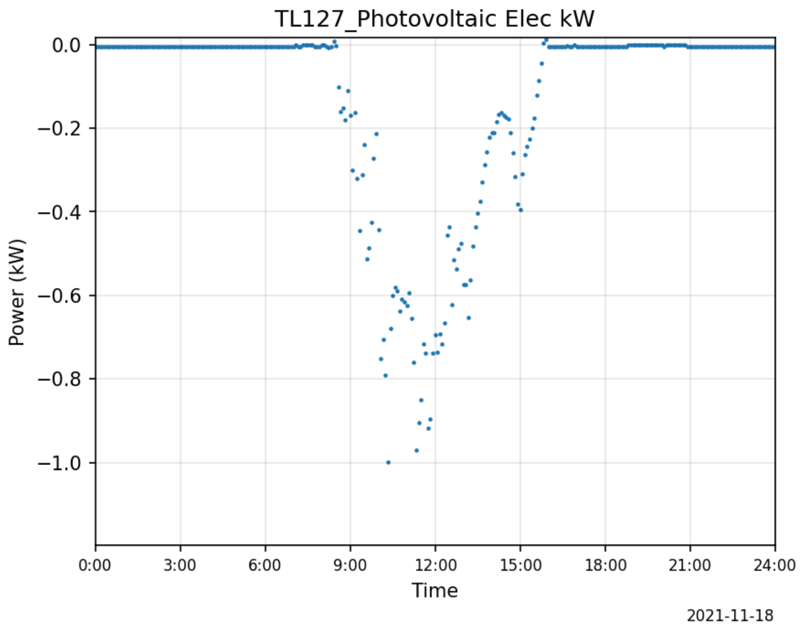
The following plots have been generated using the responses from the Enteliweb API. The graphs correspond to data measurements made by sensors set up in the RDC centre. The values span across 1 day (Friday, November 18th 2021), it gives a good indication of the general daily profile of each of these sensors.




TL102_ESB Kw Demand TL
20
15
10
5
0
0:00
3:00
6:00
9:00
12:00
15:00
18:00
21:00
24:00
Time
2021-11-18


TL125_CHP Elec kW
Power (kW)
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
0:00
3:00
6:00
9:00
12:00
15:00
18:00
21:00
24:00
Time
2021-11-18





TREND LOG TABLE CODES

Trend Log Number	Label	Path
trend-log,1	CHP_Flow Temp TL1	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,1
trend-log,2	CHP_Retn Temp TL2	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,2
trend-log,3	Boiler_Flow Temp TL3	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,3
trend-log,4	Boiler_Retn Temp TL4	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,4
trend-log,5	HeatPump_Flow Temp TL5	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,5

trend-log,6	HeatPump_Retn Temp TL6	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,6
trend-log,7	Stove_Flow Temp TL7	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,7
trend-log,8	Stove_Retn Temp TL8	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,8
trend-log,9	Buffer_Tank Temp 1 TL9	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,9
trend-log,10	Buffer_Tank Temp 2 TL10	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,10
trend-log,11	Buffer_Flow Temp TL11	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,11
trend-log,12	Buffer_Retn Temp TL12	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,12
trend-log,13	LPHW Header_Flow Temp TL13	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,13
trend-log,14	LPHW Header_Retn Temp TL14	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,14
trend-log,15	Ground_Rad Flow Temp TL15	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,15
trend-log,16	Ground_Rad Retn Temp TL16	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,16
trend-log,17	First_Rad Flow Temp TL17	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,17
trend-log,18	First_Rads Retn Temp TL18	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,18
trend-log,19	Outside Temp TL19	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,19
trend-log,20	Room_Temp_Gnd 1 TL20	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,20
trend-log,21	Room_Temp_Gnd 2 TL21	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,21
trend-log,22	Room_Temp_Gnd 3 TL22	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,22
trend-log,23	Room_Temp_Gnd 4 TL23	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,23
trend-log,24	Room_Temp_Gnd 5 TL24	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,24
trend-log,25	Room_Temp_Gnd 6 TL25	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,25
trend-log,26	Room_Temp_Gnd 7 TL26	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,26
trend-log,27	Room_Temp_Gnd 8 TL27	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,27
trend-log,28	Room_Temp_Gnd 9 TL28	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,28
trend-log,29	Room_Temp_Gnd 10 TL29	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,29
trend-log,30	Room_Co2 1 TL30	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,30
trend-log,31	Room_Co2 2 TL31	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,31
trend-log,32	Room_Co2 3 TL32	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,32
trend-log,33	Room_Co2 4 TL33	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,33
trend-log,34	Room_Co2 5 TL34	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,34
trend-log,35	Room_TVOC 1 TL35	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,35
trend-log,36	Room_TVOC 2 TL36	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,36
trend-log,37	Room_TVOC 3 TL37	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,37
trend-log,38	Room_TVOC 4 TL38	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,38
trend-log,39	Room_TVOC 5 TL39	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,39
trend-log,40	Room_Temp_Gnd 11 TL40	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,40

trend-log,41	Room_Temp_Gnd 12 TL41	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,41
trend-log,42	Room_Temp_Gnd 13 TL42	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,42
trend-log,43	Max WindSpeed SP TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,43
trend-log,44	Room_Temp_1st 15 TL44	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,44
trend-log,45	Room_Temp_1st 16 TL45	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,45
trend-log,46	Room_Temp_1st 17 TL46	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,46
trend-log,47	Room_Temp_1st 18 TL47	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,47
trend-log,48	Room_Temp_1st 19 TL48	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,48
trend-log,49	Room_Temp_1st 20 TL49	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,49
trend-log,50	Room_Temp_1st 21 TL50	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,50
trend-log,51	Room_Temp_1st 22 TL51	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,51
trend-log,52	Room_Temp_1st 23 TL52	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,52
trend-log,53	Room_Temp_1st 24 TL53	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,53
trend-log,54	Room_Temp_1st 25 TL54	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,54
trend-log,55	Room_Temp_1st 26 TL55	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,55
trend-log,56	Room_Temp_1st 27 TL56	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,56
trend-log,58	WS_Baromic Pressure TL58	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,58
trend-log,59	WS_Rain Intensity TL59	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,59
trend-log,60	WS_Outside Temp TL60	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,60
trend-log,61	WS_Outside Humidity TL61	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,61
trend-log,62	WS_Wind Direction TL62	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,62
trend-log,63	WS_Wind Speed TL63	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,63
trend-log,64	Heating Hold Off TL64	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,64
trend-log,65	Fabric Frost SP TL65	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,65
trend-log,66	Ground_Highest Temp TL66	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,66
trend-log,67	Ground_Average Temp TL67	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,67
trend-log,68	Ground_Lowest Temp TL68	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,68
trend-log,69	First_Highest Temp TL69	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,69
trend-log,70	First_Average Temp TL70	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,70
trend-log,71	First_Lowest Temp TL71	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,71
trend-log,72	Buffer_Tank Avg Temp TL72	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,72
trend-log,73	Buffer_Tank SP TL73	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,73
trend-log,74	LPHW Header SP TL74	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,74
trend-log,75	Ground_Rad Max SP TL75	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,75
trend-log,76	Ground_Rad Min SP TL76	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,76

trend-log,77	Ground_Rad Calc SP TL77	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,77
trend-log,78	First_Rad Max SP TL78	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,78
trend-log,79	First_Rad Min SP TL79	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,79
trend-log,80	First_Rad Calc SP TL80	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,80
trend-log,81	HeatPump HM_Energy TL81	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,81
trend-log,82	CHP HM_Energy TL82	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,82
trend-log,83	Boiler_Energy TL83	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,83
trend-log,84	Solar_Energy TL84	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,84
trend-log,85	Ground_Energy TL85	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,85
trend-log,86	First_Energy TL86	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,86
trend-log,87	ThermalStore_Energy TL87	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,87
trend-log,88	HeatPump HM_Power TL88	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,88
trend-log,89	CHP HM_Power TL89	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,89
trend-log,90	Boiler HM_Power TL90	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,90
trend-log,91	Solar HM_Power TL91	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,91
trend-log,92	Ground HM_Power TL92	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,92
trend-log,93	First HM_Power TL93	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,93
trend-log,94	ThermalStore HM_Power TL94	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,94
trend-log,95	Main Gas Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,95
trend-log,96	Boiler Gas Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,96
trend-log,97	CHP Gas Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,97
trend-log,98	Main Water Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,98
trend-log,99	CWS Water Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,99
trend-log,100	HWS Water Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,100
trend-log,101	Rainwater CWS Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,101
trend-log,102	Rainwater Irrigation Meter TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,102
trend-log,103	ESB kWh TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,103
trend-log,104	ESB Kw Demand TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,104
trend-log,105	Kitchen Elec kW TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,105
trend-log,106	Kitchen Elec kWh TL	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,106
trend-log,107	Bin Store Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,107
trend-log,108	Bin Store Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,108
trend-log,109	Container Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,109
trend-log,110	Container Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,110
trend-log,111	FAP Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,111

trend-log,112	FAP Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,112
trend-log,113	RWHP Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,113
trend-log,114	RWHP Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,114
trend-log,115	Comms Panel Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,115
trend-log,116	Comms Panel Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,116
trend-log,119	Distribution Board Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,119
trend-log,120	Distribution Board Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,120
trend-log,123	Workshop Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,123
trend-log,124	Workshop Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,124
trend-log,125	MCC Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,125
trend-log,126	MCC Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,126
trend-log,127	Platform Lift Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,127
trend-log,128	Platform Lift Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,128
trend-log,129	EXL Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,129
trend-log,130	EXL Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,130
trend-log,131	CHP Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,131
trend-log,132	CHP Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,132
trend-log,133	Photovoltaic Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,133
trend-log,134	Photovoltaic Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,134
trend-log,135	Lighting Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,135
trend-log,136	Lighting Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,136
trend-log,137	General Services Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,137
trend-log,138	General Services Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,138
trend-log,139	Power Elec kW	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,139
trend-log,140	Power Elec kWh	http://localhost/enteliweb/api/.bacnet/Ballymun_BH/2000/trend-log,140